GEORGIA INSTITUTE OF TECHNOLOGY | OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[X] ORIGINAL
[ ] REVISION NO. ____

Project No. E-24-637 | DATE: 8/17/81

Project Director: Dr. ~~W. B. Rouse~~ J. M. HAMMER | School/~~Lab~~ ISyE

Sponsor: NASA-Ames Research Center; Moffett Field, CA 94035

Type Agreement: Grant No. NAG 2-123

Award Period: From 8/21/81 To ~~11/20/81~~ ~~3/31/83~~ ~~9/30/82~~ ~~10/31/82~~ ~~11/30/82~~ ~~8/31/83~~ ~~9/31/84~~ ~~11-30-84~~ ~~8-30-85~~ ~~9-30-85~~ ~~9-30-86~~ 10-31-86 (Performance) ~~12/20/81~~ ~~4/30/83~~ ~~8/31/83~~ 8/31/84 (Reports)

Sponsor Amount: $30,000 | Contracted through:

Cost Sharing: * (E-24-351) | GTRI/~~GIT~~

Title: Pilot Interaction with Automated Airborne Decision Making Systems

---

ADMINISTRATIVE DATA | OCA CONTACT Duane Hutchison X4820

1) Sponsor Technical Contact: Mr. E. A. Palmer, Technical Officer; Man-Vehicle Systems Research Division, 239-3; NASA-Ames Research Center; Moffett Field, CA 94035 415/965-6147

2) Sponsor Admin./Contractual Contact: K. A. Tomberlin, Grants Officer; University Affairs Office, 241-25; NASA-Ames Research Center; Moffett Field, CA 94035 415/965-5802

Reports: See Deliverable Schedule | Security Classification: None

Defense Priority Rating: None

RESTRICTIONS

See Attached NASA Supplemental Information Sheet for Additional Requireme[nts]

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with GIT; however, the Government reserves the right to transfer title to the Government for items costing $1,000 or more.

COMMENTS: * Cost-sharing will be budgeted when the full funding is authorized.

COPIES TO:

Administrative Coordinator	Research Security Services	EES Research Public Relatio[ns]
Research Property Management	Reports Coordinator (OCA)	Project File (OCA)
Accounting Office	Legal Services (OCA)	Other: ______
Procurement/~~EES Supply Services~~	Library, Technical Reports	

1-1-74 SR7 SRS

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Date 7/31/89

roject No. E-24-637 Center No. R5231-0A0

roject Director J. M. Hammer School/Lab ISyE

ponsor NASA

ontract/Grant No. NAG 2-123 GTRC XX GIT ___

rime Contract No. N/A

itle Pilot Interaction with Automated Airborne Decision Making Systems

ffective Completion Date 12/31/87 (Performance) 12/31/87 (Reports)

loseout Actions Required:

- ___ None
- ___ Final Invoice or Copy of Last Invoice
- X Final Report of Inventions and/or Subcontracts-Questionnaire sent to PI
- ___ Government Property Inventory & Related Certificate
- ___ Classified Material Certificate
- ___ Release and Assignment
- ___ Other ___

ncludes Subproject No(s). E-16-693/ Sankar/ AE (Already closed out)

ubproject Under Main Project No. ___

ontinues Project No. ___ Continued by Project No. ___

istribution:

- X Project Director
- X Administrative Network
- X Accounting
- X Procurement/GTRI Supply Services
- X Research Property Management
- ___ Research Security Services
- ___ Reports Coordinator (OCA)
- X GTRC
- X Project File
- 2 Contract Support Division (OCA)
- ___ Other ___

E-24-637

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

February 1982 - August 1982

William B. Rouse, Principal Investigator

John M. Hammer

Ruston M. Hunt

Nancy M. Morris

Janet L. Fath

Maureen A. Reidy

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Futher, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

THE COMPUTER-AIDED COCKPIT

After several years of investigation of various aiding techniques, it was decided to take an overall, "top-down" view of the problem. Thus far, an article has been written [Hammer and Rouse, 1982] that describes the advanced cockpit, and based on this article is an initial programming project that is designed to "follow" the flight crew and aircraft through various flight phases, procedures, and procedure steps.

The data being used to develop and test this program comes from an experiment under a previous grant (NSG 2119). In this experiment, two-person crews flew a twin-engine aircraft simulator under normal, emergency, and double emergency situations [Rouse, et al., 1982]. The data for a flight is a stream of elements of the aircraft state that were recorded at any significant change and are accompanied by timestamps.

The program reads this data stream, stores it in a database, and interprets it in terms of a hierarchical, procedural model of flight. This model is a tree with a top node labeled "flight." Beneath the flight node are nodes representing the various phases of flight: preflight, taxi, postflight, etc. Beneath each phase node are the procedure nodes, which correspond to flight crew procedures such as engine run-up, pre-takeoff checklist, etc. At the lowest level are procedure steps (e.g., setting mixtures, propellers, and throttles).

The hierarchical, procedural model, as described so far, is static. The dynamic element, which observes the aircraft state database, uses routines that are unique to every node. Currently, three routines are used and each determines whether the node: (1) can be done (feasibility), (2) is being done (the flight crew's current action), or (3) has been finished (is now complete or has been completed since last checked). Each node routine can be a completely general program, though the typical program statements test current and recent past values in the database and examine the status of nearby nodes (at a higher or lower level of detail) in the tree. Tracking the flight consists of invoking these routines to determine the state of each node.

The current status of this program is that the hierarchical structure is done and that node routines are being coded. When completed, the program will be tested and refined using one half the data runs collected in the previous experiment. The remaining one half of the data will be used to validate the program.

Concurrently with the above software efforts, work has progressed on redesign of the Center's DC-8 simulator. Most of the conventional instrumentation has been removed from the pilots' panels and, at this point in time, a single CRT installed in the left panel. Some Apple II-generated displays have been used for demonstration purposes. Current plans involve expanding to multiple CRTs driven by a VAX-11/780.

STUDIES OF HUMAN PROBLEM SOLVING

All of the computer aiding schemes that have been devised throughout the course of this grant have focused on human decision making and problem solving. The design of such aids should be based on knowledge of human behavior in decision making and problem solving situations. This section reviews current efforts aimed at increasing knowledge in this area.

On of the difficult aspects of problem solving in the aircraft domain is the time-varying nature of problems in dynamic systems. The situation is also made difficult by the fact that the crew must keep the aircraft flying while also trying to solve the problem of interest. Very little research has been done in the area of humans' abilities to coordinate problem solving with ongoing operations. Nevertheless, at least one major aircraft accident has been attributed to precisely this problem [NTSB, 1973].

To investigate human problem solving in dynamic environments, a simulation called PLANT (Production Levels and Network Troubleshooting) was developed. PLANT requires subjects to configure and bring a system up to normal operating conditions and subsequently monitor these conditions. During both the transition and steady-state operations, failures can occur and subjects must diagnose these failures while also maintaining operations. In 1981, under a previous grant (NSG-2119), two experiments were conducted using PLANT. The most important result was that subjects appeared to focus on failures to the detriment of overall operations [Rouse and Morris, 1981a,b].

During the current reporting period, two efforts have been completed and one initiated in this area. First, PLANT was extended to be more realistic and was reprogrammed to utilize the color graphics of an Apple II computer. The changes made to increase realism included higher-fidelity second-order state equations for each pair of tanks, a safety system that could "trip" valves and pumps, and both valve and pump failures.

An experiment was performed to evaluate human performance using this more realistic PLANT [Fath, 1982]. The independent variables included network size (9 or 25 tanks), dynamics (slow or fast response), and failure rate (low or high). The results showed that performance was systematically affected by these variables. These results also indicated that two changes were needed: 1) the penalty for ignoring failures was not substantial enough, 2) the fast dynamics were very difficult to control over a long period of time. These conclusions are resulting in changes of PLANT for the next experiment (see below).

The second effort completed in this area during the current reporting period was an initial attempt effort to model human behavior in controlling PLANT [Reidy, 1982]. Two models were developed, one for the older version of PLANT and one for the newer version of PLANT. The models are descriptive in the sense that they were limited to using human-like strategies, but are also normative in the sense that their parameters were adjusted to achieve the best performance possible for the strategy chosen.

This procedure was used to allow evaluation of human performance relative to a realistic optimal.

Results of this evaluation indicated that the more difficult levels of the independent variables (e.g., larger networks or faster dynamics) were more troublesome for subjects than for the models. In particular, the earlier notion [Rouse and Morris, 1981a,b] that an increased frequency of failures causes subjects to decrease production more than necessary was supported. Further, as was found experimentally for subjects [Fath, 1982], the faster dynamics were difficult for the model to maintain within limits for any extended period of time.

The main purpose of the above two efforts was to assure that PLANT was a reasonable experimental environment in the sense that experimental manipulations of PLANT's parameters (e.g., network size, tank sizes, pipe sizes and lengths, etc.) would provide a range of realistic problem solving situations that are suitable for investigating fundamental issues related to human problem solving in dynamic systems.

While the issue of coordinating problem solving and operations motivated the developments reported in this section, the research has now come to focus on a more specific issue. Namely, how much does an operator need to know about his system in order to be able to cope successfully with failures? Further, does the answer to this question depend on the nature of the failures (i.e., familiar vs. unfamiliar failures)? This issue is the focus of an experiment that has been planned and will soon begin [Morris, 1982, Morris and Rouse, 1982].

For this experiment, subjects will be trained with one of three types of instructions. One set of instructions (minimal) will simply explain the goal and the set of alternatives available for achieving this goal. Another set of instructions (procedures) will provide written procedures, similar to those used in aircraft, for dealing with the types of events that will occur during training. The third set of instructions (principles) will explain the physical nature of PLANT in terms of a set of heuristics (e.g., flow rate is proportional to pressure differences) and use these heuristics to indicate why procedures work and their limits of applicability.

After training, subjects will operate PLANT in situations involving both familiar failures (i.e., those practiced during training) and unfamiliar failures which they were told could occur but in fact they never experienced during training. It is hypothesized that the procedures group may be somewhat better than the principles group for familiar failures, but the principles group will be very much better than the procedures group for unfamiliar failures. The minimal group, which will mainly serve as a control group, is expected to perform poorly relative to the other two groups for both familiar and unfamiliar failures.

If the above hypotheses are supported by the experimental results, the most important implications for this project in terms of computer aiding will be related to the level of understanding the human must have about how the aiding works in order to assume control when the unusual occurs. It is planned that this issue of the relationship between understanding and aiding will be greatly refined, and hopefully at least partially resolved, as the program of research supported by this grant proceeds.

REFERENCES

1. Fath, J.L. 1982. Design and Evaluation of a Process Plant Simulator. Atlanta, GA: Georgia Institute of Technology, Center for Man-Machine Systems Research, Technical Report No. 82-3, September.

2. Hammer, J.M., and Rouse W.B. 1982. "Design of An Intelligent Computer-Aided Cockpit," Proceedings of the 1982 IEEE International Conference on Cybernetics and Society, Seattle, October.

3. Morris, N.M. 1982. Human Problem Solving in Process Control. Ph.D. thesis in progress.

4. Morris, N.M., and Rouse, W.B. 1982. "An Approach to Assessing the Effects of Level of Understanding Upon Operator's Behavior," Proceedings of the 1982 IEEE International Conference on Cybernetics and Society, Seattle, October.

5. NTSB, 1973. Aircraft Accident Report, Eastern Airlines Inc., L-1011, N310EA, Miami, Florida, December 29, 1972. Washington, DC: National Transportation Safety Board, June 1973, NTIS No. PB222 359.

6. Reidy, M.A. 1982. Model-Based Analyses of Human Performance in Two Process Control Tasks. Atlanta, GA: Georgia Institute of Technology, Center for Man-Machine Systems Research, Technical Report No. 82-4, September.

7. Rouse, W.B., and Morris, N.M. 1982a. "Human Problem Solving in a Process Control Task," Proceedings of the First Annual European Conference on Human Decision Making and Manual Control, Delft, The Netherlands, May.

8. Rouse, W.B., and Morris, N.M. 1982b. "Studies of Human Problem Solving in a Process Control Task," Proceedings of the 1981 IEEE International Conference on Cybernetics and Society, Atlanta, October.

9. Rouse, S.H., Rouse, W.B., and Hammer, J.M., 1982. "Design and Evaluation of an Onboard Computer-Based Information System for Aircraft," IEEE Trans. on Sys., Man, and Cybernetics, Vol. SMC-12, No. 4, July/August, pp. 451-463.

E-... ...-1

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
ATLANTA, GEORGIA 30332 (404) 894-3996

March 31, 1983

NASA Scientific and Technical
Information Facility
Post Office Box 8757
Baltimore/Washington International Airport
Baltimore, MD 21240

Gentlemen:

Enclosed please find two copies of the Semiannual Report for NASA Grant No. NAG-2-123 for the period of September 1982 - February 1983. The copies include the original plus one photocopy.

Sincerely,

William B. Rouse
Professor and Director

WBR:j

cc: R. Curry
E. Palmer
University Affairs Office
F. Cochran

Encl.

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

September 1982 - February 1983

William B. Rouse, Principal Investigator

John M. Hammer

Edward N. Brown

Nancy M. Morris

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Futher, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering mehtods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments as affected by the human's level of knowledge of system operations. Both of these efforts are producing results that are planned to be tested further in the Center's new full-scale simulation facility. Progress on developing this facility is discussed in the final section of this progress report.

THE INTELLIGENT COCKPIT

The error-detecting capability of the procedural model of flight is discussed in this section. The computer program, input data, results, and future plans are presented.

The Computer Program

The computer program matches the dynamic aircraft state with a script. A script is a sequence of actions (termed elements) that are expected to occur in a particular situation. The aircraft script contains four levels. At the top level is a single element for the entire flight. Beneath it are several elements for different phases of flight (e.g., preflight, takeoff). Beneath each phase element are the procedures contained in that phase (e.g., preflight checklist, starting engines, engine run-up). Beneath each procedure element are the steps of the procedure (e.g., engage left starter).

Each element, regardless of its level in the hierarchy, is represented uniformly. Included in this representation are three rules that determine when the element can possibly be executed, is done with execution, or has aborted its execution. These rules typically check the current simulator state. The computer program tracks the flight by evaluating the three rules in the elements of the script. For example, the element "engage left starter" has a done rule that examines the simulator state variable that is connected to the left starter. When the starter is engaged, this corresponding step in the procedure is marked done.

The Input Data

The input data are from a previous experiment [Rouse, Rouse, and Hammer 1982]. A Link GAT-II simulator (twin piston engine, general aviation) was interfaced to a PDP-11/40. Approximately 75 signals were recorded. The simulated flight was direct between two airports. The simulator was flown IFR. ATC provided headings for all takeoffs and landings. Data from thirty-six flights is available; sixteen normal, sixteen emergency, and four double emergency. Testing has thus far been limited to the normal flights.

The program is being tested by equally and randomly dividing these flights into derivation and validation data. The program is developed on the derivation data so that it is able to find all of the errors previously reported in [Rouse and Rouse, 1983]. The program will be held fixed during runs with the validation data.

Results

For the seven flights in the derivation data, the program found all errors that had been previously found. Some examples of these errors are omitted procedures, inappropriate activation of various lights and anti-icing equipment, and lowering the landing gear while airspeed was too high. Furthermore, the program identified what appear to be several additional errors. Since some of these errors may be a matter of pilot preference. These questions will be resolved by consulting subject matter experts.

Though the program's detection of errors is an easily measured goal, it is really a by-product. The principal goal is to keep the program's internal procedural model in agreement with what the flight crew and the aircraft are doing. Eventually, this will be measured by the latency between the actual event and the time when the model was so updated. Short latency is necessary if the procedural model is to be of any value as an online decision aid. For example, the model will be of little use if it could only detect some errors 30 seconds after they were committed.

Future Plans

There are two goals to be pursued in the near future. The most immediate goal is to analyze the remaining validation data. Second, the program script will be enlarged for emergency procedures, and flights with emergencies will be analyzed for errors. Emergency situations are, of course, a more important area for aiding flight crew decision making.

References

1. Rouse, S.H., Rouse, W.B., and Hammer, J.M., "Design and Evaluation of an Onboard Computer-Based Information System for Aircraft," IEEE Trans. Systems, Man, and Cybernetics, Vol. 12, No. 4, 1982.

2. Rouse, W.B. and Rouse, S.H., "Analysis and Classification of Human Error," to appear in IEEE Trans. Systems, Man, and Cybernetics.

STUDIES OF HUMAN PROBLEM SOLVING

The topic of human problem solving and decision making with respect to dynamic systems continues to be of interest to this research group. The current question under investigation is this: How much and what kind of system-relevant information does the human operator need in order to control the system effectively and correct problem situations when they occur? This question is being investigated in the context of a dynamic simulation called PLANT (Production Levels And Network Troubleshooting), which was discussed in earlier reports.

The human PLANT operator is required to configure and bring the system up to normal operating conditions, monitor the system for malfunctions, and take steps to correct and recover from failures if any should occur. The results of experimental efforts using PLANT have thus far indicated that the PLANT environment is a reasonable one for presenting problem solving situations suitable for investigation of human problem solving behavior in dynamic systems.

In the current experiment, the parameters of the PLANT (e.g., tank and pipe sizes, network size, etc.) are held constant within and across subjects. The major variables of interest do not involve the effects of system characteristics upon operator performance (as was the case in previous experiments), but are rather concerned with the nature and extent of system-relevant knowledge that operators have, and the relationships between this knowledge and performance. Experimental subjects control a relatively "slow" PLANT (in that the dynamics of the system are such that responses to control actions tend to be "sluggish"), with 9 tanks in the system.

In an effort to manipulate the PLANT-relevant knowledge that subjects have, instructions to subjects are varied in a between-groups design. One group receives a minimal set of instructions, which merely informs them of the goals of PLANT operation, the types of failures which may occur, and the command options available to them as operators. Another group is provided information about the behavior of the system in terms of dynamic relationships (principles), in addition to the minimal instructions. A third group is provided both the minimal instructions and a set of operational heuristics and procedures, which are designed to help the operator achieve the goals of PLANT operation. Finally, a fourth group is provided with all of the PLANT information described here. The system-relevant knowledge a subject actually possesses is evaluated with a multiple-choice test administered at the end of the last experimental session.

Following a training period, all subjects operate the PLANT in situations involving both familiar failures (i.e., those experienced during training) and unfamiliar failures (i.e., failures which they are told may happen but which never occur during training). It is hypothesized that those groups receiving either principles or procedures will perform equally well in familiar situations, but that the groups receiving principles will be better than those groups without dynamics in unfamiliar situations. The group receiving only minimal instructions is expected to perform less well than the other groups in both familiar and unfamiliar situations.

As this report is written, approximately half of the data have been collected, with groups 1 and 4 (minimal and principles + procedures) completed. A preliminary analysis of the data yields interesting results.

The most obvious difference between the two groups completed thus far has been their performance on the multiple-choice test administered at the end of the experiment. The group receiving minimal instructions scored an average of 65% correct, as opposed to a mean correct of 84% achieved by the full (principles + procedures) instruction group. This difference was statistically significant ($p < .01$). The difference in test scores is not surprising, in light of the fact that the test was designed specifically to test knowledge gained from the instructions provided. It is noteworthy, however, that the scores do not reflect what might have been expected if PLANT-relevant knowledge were obtained *only* from the written instructions. For instance, the group receiving minimal instructions might have been expected to answer slightly more than 35% of the test questions correctly, whereas the group receiving full instructions should have been able to respond correctly to almost all of the questions. Obviously the minimal instruction group learned a great deal about PLANT via experience with the system; apparently the full instruction group did not learn all that the instructions contained.

A number of other dependent variables have been examined, which may be broadly classified as either "global" measures or "fine-grained" measures. The global measures indicate how well subjects were able to achieve the goals emphasized in the instructions: to produce as much as possible, and keep safety system trips at a minimum. Fine-grained measures might be called "process" measures, in that they reveal information about the behaviors exhibited by subjects as they attempted to achieve their goals.

Two of the global measures are of interest: production achieved, and frequency of safety system trips. There was no difference between the two groups in the amount of production achieved. Although the full instruction group produced approximately 5% more, this difference was not statistically significant; there was, however, greater variability between subjects in the minimal instruction group. Significant differences in production were found to be related to the familiarity of the failures encountered, with production during familiar situations exceeding production under unfamiliar circumstances by approximately 8%. The two groups did differ significantly in the number of safety system trips which occurred during their production runs; whereas the groups were able to achieve approximately the same production, the minimal instruction group had 60% more trips than did the full instruction group (416.9 vs. 260.4).

The fine-grained measures paint a clearer picture of the differences between the two groups. In general, the control performance of the minimal instruction group was less "optimal" and less consistent than that of the full instruction group. For example, the minimal instruction group kept fewer valves open on the average (even though both groups were told that the PLANT was capable of producing more product if all valves were open), and had greater differences in fluid levels within the system.

It is interesting to note the differences between the two groups in their response to unfamiliar situations. One of the unfamiliar situations was apparently so evident to both groups that its occurrence had no effect upon performance. The other unfamiliar situation was more subtle, though, and had an interesting effect upon subjects' control performance. Whereas both groups were fairly consistent across all situations in the average amount of output that was specified, there were differences in the specified inputs. The full instruction group's response to the unfamiliar situation was to slightly reduce the amount of fluid input into the system, with no change in the variability of inputs specified. The minimal instruction group reacted in just the opposite manner: the average amount of input was greatly increased, as was the variability of input specifications. In other words, the full instruction group did not dramatically change their control strategy, but became slightly more conservative; the minimal instruction group faced with the unfamiliar situation maintained system output but was forced to adopt a "bang-bang" approach to system input in order to maintain an acceptable level of fluid within the system.

It should be noted that there were no differences between groups in their detection of both familiar and unfamiliar failures. The average time to repair familiar failures was approximately equal for the two groups, and there was no difference in detection of the unfamiliar failures. It is also interesting to note that there was often a significant effect of session upon the dependent variables analyzed. At the end of the experiment the performance of both groups was still improving; at this stage in the analysis, it appears that the improvement of the two groups was essentially parallel, with both groups improving at the same rate.

Data collection in the current experiment is expected to continue until mid-May, at which time the four groups described will have been completed. At that time it should be possible to more fully describe the effects of instructions and system knowledge upon performance. The observations reported here will be reexamined in light of the new data, and more definite conclusions will be drawn.

FLIGHT SIMULATOR FACILITY

A DC-8 simulator is being modernized to provide a realistic testbed for experiments in decision aiding. The original electromechanical instrumentation was removed and high resolution, color CRT monitors with alphanumeric keyboards are being installed in place. Existing flight dynamics software is being modified to run on the Center's VAX-11/780.

Two high resolution 13" color monitors will be installed, one each directly in front of the pilot and copilot. These monitors will display graphical information simulating an HSI and ADI. They will be driven, at least initially, by two existing Apple II microcomputers.

Two 9" b/w monitors will be installed between the color monitors. These b/w monitors will display both text and low resolution graphics, such as procedure checklists and engine instruments. They will be driven by a composite video signal tapped from inside an existing computer terminal.

Two 5" b/w monitors will be installed in the pedestal. Due to their small size, they will be used for menus. They will be driven by signals tapped from computer terminals. The keyboards in conjunction with the 5" monitors will be the primary pilot interface with the computer system.

The type of keyboard to be installed was originally used in the experiments run on the Link GAT-II simulator at the University of Illinois. It is relatively small (4.5 by 9 inches) and has 60 keys--enough for all A-Z, 0-9, and some dedicated function keys.

Flight dynamics and display software have been obtained from NASA. This software was developed by Lockheed-Georgia under contract to NASA. It runs on VAX and SEL processors with Ikonas graphics. This package will be modified to produce a DC-8 flight dynamics simulation and a display system capable of working with simpler Apple II graphics.

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

September 1982 - February 1983

William B. Rouse, Principal Investigator

John M. Hammer

Edward N. Brown

Nancy M. Morris

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Futher, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering mehtods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments as affected by the human's level of knowledge of system operations. Both of these efforts are producing results that are planned to be tested further in the Center's new full-scale simulation facility. Progress on developing this facility is discussed in the final section of this progress report.

THE INTELLIGENT COCKPIT

The error-detecting capability of the procedural model of flight is discussed in this section. The computer program, input data, results, and future plans are presented.

The Computer Program

The computer program matches the dynamic aircraft state with a script. A script is a sequence of actions (termed elements) that are expected to occur in a particular situation. The aircraft script contains four levels. At the top level is a single element for the entire flight. Beneath it are several elements for different phases of flight (e.g., preflight, takeoff). Beneath each phase element are the procedures contained in that phase (e.g., preflight checklist, starting engines, engine run-up). Beneath each procedure element are the steps of the procedure (e.g., engage left starter).

Each element, regardless of its level in the hierarchy, is represented uniformly. Included in this representation are three rules that determine when the element can possibly be executed, is done with execution, or has aborted its execution. These rules typically check the current simulator state. The computer program tracks the flight by evaluating the three rules in the elements of the script. For example, the element "engage left starter" has a done rule that examines the simulator state variable that is connected to the left starter. When the starter is engaged, this corresponding step in the procedure is marked done.

The Input Data

The input data are from a previous experiment [Rouse, Rouse, and Hammer 1982]. A Link GAT-II simulator (twin piston engine, general aviation) was interfaced to a PDP-11/40. Approximately 75 signals were recorded. The simulated flight was direct between two airports. The simulator was flown IFR. ATC provided headings for all takeoffs and landings. Data from thirty-six flights is available; sixteen normal, sixteen emergency, and four double emergency. Testing has thus far been limited to the normal flights.

The program is being tested by equally and randomly dividing these flights into derivation and validation data. The program is developed on the derivation data so that it is able to find all of the errors previously reported in [Rouse and Rouse, 1983]. The program will be held fixed during runs with the validation data.

Results

For the seven flights in the derivation data, the program found all errors that had been previously found. Some examples of these errors are omitted procedures, inappropriate activation of various lights and anti-icing equipment, and lowering the landing gear while airspeed was too high. Furthermore, the program identified what appear to be several additional errors. Since some of these errors may be a matter of pilot preference. These questions will be resolved by consulting subject matter experts.

Though the program's detection of errors is an easily measured goal, it is really a by-product. The principal goal is to keep the program's internal procedural model in agreement with what the flight crew and the aircraft are doing. Eventually, this will be measured by the latency between the actual event and the time when the model was so updated. Short latency is necessary if the procedural model is to be of any value as an online decision aid. For example, the model will be of little use if it could only detect some errors 30 seconds after they were committed.

Future Plans

There are two goals to be pursued in the near future. The most immediate goal is to analyze the remaining validation data. Second, the program script will be enlarged for emergency procedures, and flights with emergencies will be analyzed for errors. Emergency situations are, of course, a more important area for aiding flight crew decision making.

References

1. Rouse, S.H., Rouse, W.B., and Hammer, J.M., "Design and Evaluation of an Onboard Computer-Based Information System for Aircraft," IEEE Trans. Systems, Man, and Cybernetics, Vol. 12, No. 4, 1982.

2. Rouse, W.B. and Rouse, S.H., "Analysis and Classification of Human Error," to appear in IEEE Trans. Systems, Man, and Cybernetics.

STUDIES OF HUMAN PROBLEM SOLVING

The topic of human problem solving and decision making with respect to dynamic systems continues to be of interest to this research group. The current question under investigation is this: How much and what kind of system-relevant information does the human operator need in order to control the system effectively and correct problem situations when they occur? This question is being investigated in the context of a dynamic simulation called PLANT (Production Levels And Network Troubleshooting), which was discussed in earlier reports.

The human PLANT operator is required to configure and bring the system up to normal operating conditions, monitor the system for malfunctions, and take steps to correct and recover from failures if any should occur. The results of experimental efforts using PLANT have thus far indicated that the PLANT environment is a reasonable one for presenting problem solving situations suitable for investigation of human problem solving behavior in dynamic systems.

In the current experiment, the parameters of the PLANT (e.g., tank and pipe sizes, network size, etc.) are held constant within and across subjects. The major variables of interest do not involve the effects of system characteristics upon operator performance (as was the case in previous experiments), but are rather concerned with the nature and extent of system-relevant knowledge that operators have, and the relationships between this knowledge and performance. Experimental subjects control a relatively "slow" PLANT (in that the dynamics of the system are such that responses to control actions tend to be "sluggish"), with 9 tanks in the system.

In an effort to manipulate the PLANT-relevant knowledge that subjects have, instructions to subjects are varied in a between-groups design. One group receives a minimal set of instructions, which merely informs them of the goals of PLANT operation, the types of failures which may occur, and the command options available to them as operators. Another group is provided information about the behavior of the system in terms of dynamic relationships (principles), in addition to the minimal instructions. A third group is provided both the minimal instructions and a set of operational heuristics and procedures, which are designed to help the operator achieve the goals of PLANT operation. Finally, a fourth group is provided with all of the PLANT information described here. The system-relevant knowledge a subject actually possesses is evaluated with a multiple-choice test administered at the end of the last experimental session.

Following a training period, all subjects operate the PLANT in situations involving both familiar failures (i.e., those experienced during training) and unfamiliar failures (i.e., failures which they are told may happen but which never occur during training). It is hypothesized that those groups receiving either principles or procedures will perform equally well in familiar situations, but that the groups receiving principles will be better than those groups without dynamics in unfamiliar situations. The group receiving only minimal instructions is expected to perform less well than the other groups in both familiar and unfamiliar situations.

As this report is written, approximately half of the data have been collected, with groups 1 and 4 (minimal and principles + procedures) completed. A preliminary analysis of the data yields interesting results.

The most obvious difference between the two groups completed thus far has been their performance on the multiple-choice test administered at the end of the experiment. The group receiving minimal instructions scored an average of 65% correct, as opposed to a mean correct of 84% achieved by the full (principles + procedures) instruction group. This difference was statistically significant ($p < .01$). The difference in test scores is not surprising, in light of the fact that the test was designed specifically to test knowledge gained from the instructions provided. It is noteworthy, however, that the scores do not reflect what might have been expected if PLANT-relevant knowledge were obtained <u>only</u> from the written instructions. For instance, the group receiving minimal instructions might have been expected to answer slightly more than 35% of the test questions correctly, whereas the group receiving full instructions should have been able to respond correctly to almost all of the questions. Obviously the minimal instruction group learned a great deal about PLANT via experience with the system; apparently the full instruction group did not learn all that the instructions contained.

A number of other dependent variables have been examined, which may be broadly classified as either "global" measures or "fine-grained" measures. The global measures indicate how well subjects were able to achieve the goals emphasized in the instructions: to produce as much as possible, and keep safety system trips at a minimum. Fine-grained measures might be called "process" measures, in that they reveal information about the behaviors exhibited by subjects as they attempted to achieve their goals.

Two of the global measures are of interest: production achieved, and frequency of safety system trips. There was no difference between the two groups in the amount of production achieved. Although the full instruction group produced approximately 5% more, this difference was not statistically significant; there was, however, greater variability between subjects in the minimal instruction group. Significant differences in production were found to be related to the familiarity of the failures encountered, with production during familiar situations exceeding production under unfamiliar circumstances by approximately 8%. The two groups did differ significantly in the number of safety system trips which occurred during their production runs; whereas the groups were able to achieve approximately the same production, the minimal instruction group had 60% more trips than did the full instruction group (416.9 vs. 260.4).

The fine-grained measures paint a clearer picture of the differences between the two groups. In general, the control performance of the minimal instruction group was less "optimal" and less consistent than that of the full instruction group. For example, the minimal instruction group kept fewer valves open on the average (even though both groups were told that the PLANT was capable of producing more product if all valves were open), and had greater differences in fluid levels within the system.

It is interesting to note the differences between the two groups in their response to unfamiliar situations. One of the unfamiliar situations was apparently so evident to both groups that its occurrence had no effect upon performance. The other unfamiliar situation was more subtle, though, and had an interesting effect upon subjects' control performance. Whereas both groups were fairly consistent across all situations in the average amount of output that was specified, there were differences in the specified inputs. The full instruction group's response to the unfamiliar situation was to slightly reduce the amount of fluid input into the system, with no change in the variability of inputs specified. The minimal instruction group reacted in just the opposite manner: the average amount of input was greatly increased, as was the variability of input specifications. In other words, the full instruction group did not dramatically change their control strategy, but became slightly more conservative; the minimal instruction group faced with the unfamiliar situation maintained system output but was forced to adopt a "bang-bang" approach to system input in order to maintain an acceptable level of fluid within the system.

It should be noted that there were no differences between groups in their detection of both familiar and unfamiliar failures. The average time to repair familiar failures was approximately equal for the two groups, and there was no difference in detection of the unfamiliar failures. It is also interesting to note that there was often a significant effect of session upon the dependent variables analyzed. At the end of the experiment the performance of both groups was still improving; at this stage in the analysis, it appears that the improvement of the two groups was essentially parallel, with both groups improving at the same rate.

Data collection in the current experiment is expected to continue until mid-May, at which time the four groups described will have been completed. At that time it should be possible to more fully describe the effects of instructions and system knowledge upon performance. The observations reported here will be reexamined in light of the new data, and more definite conclusions will be drawn.

FLIGHT SIMULATOR FACILITY

A DC-8 simulator is being modernized to provide a realistic testbed for experiments in decision aiding. The original electromechanical instrumentation was removed and high resolution, color CRT monitors with alphanumeric keyboards are being installed in place. Existing flight dynamics software is being modified to run on the Center's VAX-11/780.

Two high resolution 13" color monitors will be installed, one each directly in front of the pilot and copilot. These monitors will display graphical information simulating an HSI and ADI. They will be driven, at least initially, by two existing Apple II microcomputers.

Two 9" b/w monitors will be installed between the color monitors. These b/w monitors will display both text and low resolution graphics, such as procedure checklists and engine instruments. They will be driven by a composite video signal tapped from inside an existing computer terminal.

Two 5" b/w monitors will be installed in the pedestal. Due to their small size, they will be used for menus. They will be driven by signals tapped from computer terminals. The keyboards in conjunction with the 5" monitors will be the primary pilot interface with the computer system.

The type of keyboard to be installed was originally used in the experiments run on the Link GAT-II simulator at the University of Illinois. It is relatively small (4.5 by 9 inches) and has 60 keys--enough for all A-Z, 0-9, and some dedicated function keys.

Flight dynamics and display software have been obtained from NASA. This software was developed by Lockheed-Georgia under contract to NASA. It runs on VAX and SEL processors with Ikonas graphics. This package will be modified to produce a DC-8 flight dynamics simulation and a display system capable of working with simpler Apple II graphics.

Research Svcs.

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
ATLANTA, GEORGIA 30332
(404) 894-3996

November 30, 1983

NASA Scientific and Technical
Information Facility
Post Office Box 8757
Baltimore/Washington International Airport
Baltimore, MD 21240

Gentlemen:

Enclosed please find two copies of the Semiannual Report for NASA Grant No. NAG-2-123 for the period of March 1983 - August 1983. The copies include the original plus one photocopy.

Sincerely,

William B. Rouse
Professor and Director

cc: R. Curry
E. Palmer
University Affairs Office
F. Cochran

Enclosure

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

March 1983 - August 1983

William B. Rouse, Principal Investigator

John M. Hammer

Nancy M. Morris

Edward N. Brown

Wan C. Yoon

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Further, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering methods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments as affected by the human's level of knowledge of system operations. Both of these efforts are producing results that are planned to be tested further in the Center's new full-scale simulation facility. Progress on developing this facility is discussed in the third and final section of this progress report.

THE INTELLIGENT COCKPIT

This project is a direct descendent of work by the authors on human-computer interaction in the cockpit dating back to 1975. As this research has evolved, the modeling and analysis methods that have emerged have enabled consideration of increasingly complex domains. For example, two of the more recent sets of studies considered pilot (and crew) problem solving in full-mission simulation studies [Rouse et al., 1982; Johannsen and Rouse, 1983].

The perspectives provided by these years of research have resulted in an integrated computer aiding concept which the authors have termed the "intelligent cockpit." The overall outline of this concept is outlined in Hammer and Rouse [1982]. The basic idea is to use advanced software engineering methods (e.g., from artificial intelligence) and models of human decision making and problem solving to produce a computer-based aid that "understands" what is going on in the cockpit and can provide assistance accordingly.

This very ambitious project is being pursued in an incremental manner. The first increment is an intelligent flight management aid that understands the nature of procedures and can monitor their execution. The paper by John Hammer in the Appendix of this report summarizes this work. The results reported in this paper prove the soundness of the concepts; the next stage will be to implement this idea in the Center's simulator to allow full-scale testing.

References

1. Hammer, J.M. and Rouse, W.B. 1982. "Design of An Intelligent Computer-Aided Cockpit." Proceedings of the 1982 IEEE International Conference on Cybernetics and Society, Seattle, October, pp. 449-453.

2. Johannsen, G. and Rouse, W.B. 1983. "Studies of Planning Behavior of Aircraft Pilots in Normal, Abnormal, and Emergency Situations," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-13. No. 3, May/June 1983, pp. 267-278.

3. Rouse, S.H., Rouse, W.B. and Hammer, J.M. 1982. "Design and Evaluation of An Onboard Computer-Based Information System for Aircraft." IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 4, July/August 1982, pp. 451-463.

STUDIES OF HUMAN PROBLEM SOLVING

In order to support domain-oriented projects such as the intelligent cockpit. it is necessary to increase our basic understanding of human decision making and problem solving. This has been a main tenet of this program of research since its inception and continues to be a guiding principle.

The latest efforts in the area of basic studies of human problem solving have focused on the question of what humans need to know about a dynamic process in order to be able to deal successfully with unfamiliar and unanticipated events. The paper by Nancy Morris in the Appendix of this report summarizes the results of a study that compared knowledge of operating procedures and physical principles in a process control task. One of the most interesting results was that knowledge of physical principles, as assessed via a written test, did not result in improved performance. Of the many important issues that this raises, two of particular note are the nature of training (e.g., "what" vs. "how") and the appropriate forms for different types of knowledge.

FLIGHT SIMULATION SOFTWARE

The progress report for the last reporting period discussed the hardware modifications planned for the Center's DC-8 simulator in order to create an advanced cockpit simulator. This section focuses on software developments.

Software has been developed to produce a B-747 flight simulation on the Center's Vax 11/780. At this point in time, it employs simplified dynamics to simulate the motion of an aircraft and only a few of the essential subsystems. Despite this simplicity, the software meets our overall need to provide pilots with a reasonably realistic environment for the purpose of investigating their problem solving behavior in various situations.

The program allows the pilot to activate the control surfaces of the jet aircraft, adjust engine thrust, and tune navigational radio equipment. The program responds to commands by adjusting aircraft attitude to match the control surfaces and updating the instrument panel display as the trajectory of the aircraft evolves through space.

An instrument panel was designed to display information that comes from the flight simulation. This information is composed of current aircraft attitude, positions of switches, flight situation, and navigational environment. Included are pitch, altitude, engine thrust, compass, fuel, landing gear, brake, VOR system, stall warning, VLF OMEGA, ILS, VHF channel, etc. This brief panel gives the pilot all the basic flight information he needs during the three primary mission phases (i.e., takeoff, navigation, and landing) using standard flight procedures and radio facilities.

Although the pilot completely controls the motion of the jet, wind forces that vary with altitude can influence the flight. An analytical combination of jet and wind motion yields the true position of the jet relative to the earth's surface.

This simulation software, however, is still incomplete. The current interface--keyboard and screen--are suitable for software development but will have to change to use the simulator displays and controls. The existing single instrument panel must be rearranged into several different CRT's. The flight control will be executed by a pilot who will be sitting down in a full scale aircraft cockpit facility and using a control stick and a flight deck of high fidelity.

Also, more subsystems will be involved to cover all information that would be necessary for a realistic problem solving environment. Among these subsystems are the engines (giving engine pressure ratio, fuel flow, exhaust gas temperature, etc.), hydraulic system, autopilot, fuel, CDTI, etc. With all these subsystems, it is believed that the flight simulator will be an appropriate base for studies of aiding problem solving.

APPENDIX

AN INTELLIGENT FLIGHT MANAGEMENT AID FOR PROCEDURE EXECUTION

John M. Hammer

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

A computer program is described which contains a model of the procedures used in the operation of a twin engine aircraft. This program, by comparing the model to the aircraft state, can determine when a procedure (or checklist) should be or is invoked and when each step (detectable by a change in the aircraft state) is completed. Thus, the program tracks the flight crew's procedure execution through changes in the aircraft state. Data was used for evaluation from an earlier experiment on a Link GAT-II simulator. The program was able to identify practically all of the errors identified by hand as well as locate some that had been missed by human judges. It is felt that this model could signficantly aid flight crews.

INTRODUCTION

A computer program for detecting pilot error is described. This program observes pilot actions through the aircraft controls and state. These actions are compared to those of a procedure script, which can be considered a prescriptive model of the procedural aspects of flight. A pilot error, then, is a discrepancy between the pilot actions and the script. The program is capable of detecting omitted, incorrect, or out-of-order steps as well as certain irrelevant actions.

This procedural aid is part of a larger research thrust known as the intelligent cockpit. Its goals are to demonstrate concepts for a system capable of intelligent decision-aiding in flight management. For example, Hammer and Rouse [1982] have identified a number of levels at which aiding could occur. At the lowest level, computerized warning, calculation, and display control could, if implemented properly, improve the human factors of the cockpit. Most flight deck automation is concerned with this level [Wiener and Curry 1980]. At a second level, the computer could check that certain conditions were met and infer the intentions of the flight crew. For example, the system described later checks pilot actions against a prescribed plan and infers what procedure the crew should be following. At the highest level, the computer could compensate for and advise the pilot. Compensation could mean taking over some task that had been allocated to the pilot or correcting for pilot error. Advice could take the form of a natural language dialogue on some cooperative human-computer problem solving. Both of these forms of aiding are beyond the current state of the art.

Problem Statement

Currently, pilot error (that goes undetected by the pilot) is found by humans who examine simulator traces (as above) or cockpit voice recordings. It would be better to detect these errors seconds after they occurred while there was still time to correct them. This was the goal of the research reported here. Software was written to implement a model of procedure execution during flight. The model is continually updated (steps noted done, procedures invoked and finished) during the flight so as to keep it close to what the flight crew is doing. The model could be used as an aid since it has some understanding of when a procedure is to be used and what its execution entails. To evaluate the model, it was tested on earlier simulator flights with previously indentified errors. The figure of merit was the number of these errors that the model could locate.

The remainder of this article contains sections on previous work on procedural error, programming methodologies for procedural aiding, the pilot's procedures, the internal program organization of the aid, evaluations of the aid, and conclusions.

PREVIOUS WORK

Humans occasionally err when following procedures. The forms of error have frequently been observed to be steps not executed, done out-of-order, done incompletely, or done at the wrong time. The same is true of whole procedures, which are sometimes incorrectly selected for execution. Some theories [Norman 1981] and many classification schemes [Rasmussen 1979] [Monan 1979] [van Eekhout and Rouse 1981] have been put forth, and are reviewed in Rouse and Rouse [1983]. The theories and classification schemes will not be reviewed here, since the goal is to build an aid for reducing error, not to explain it.

The remainder of this literature review contains two parts: one on other aids for reducing procedural error, and a second on a line of research leading directly to this research.

Goodstein [1979] has proposed a computerized procedure display system. Its design was based on the belief that the operator executes procedures with some goal in mind -- changing the system via procedures from one state to another. Consequently, the system explicitly displays this hierarchy. The procedure environment is also enriched by including preconditions, constraints, and warnings along with the procedure text.

The proposed system was to be implemented with three displays. The first would display the sequence of procedures to be followed. Included in this would be the status of various procedures (e.g., on hold or waiting for the plant to respond). The second display would contain the text of a single procedure along with supplemental preconditions, constraints, and warnings. The third display is for support. It might display the relevant plant status so that the operator would not have to walk away from the displays just to read an instrument. While this proposed three display system would seem to be an improvement over current practice, it does not appear to have been evaluated with human operators.

Colley [1982] and Seeman et al. [1982] are in the process of developing a computerized procedure support system for nuclear power plant operators. The system compares the current plant state to a set of desirable, or safe, plant states. A procedure is then generated by the computer to transform the plant to the nearest safe state. A practical advantage of automatic procedure generation is that a potentially larger set of procedures could be available than would from hardcopy. For the latter, the system designers cannot afford to create every possible procedure. If a computer could derive the procedures from some general principles, automatic generation could represent a considerable improvement.

The current system can produce procedures for an eighteen component lubrication system. The procedures are generated dynamically; i.e., after each operator action, the system regenerates an appropriate procedure. Thus, if the operator errs, an appropriate change will appear in the procedure text. The development effort should be viewed as an attempt to produce a methodology for procedure generation. It has not yet been tested on operators.

Background Work Leading Directly to this Study

This section discusses a sequence of studies that lead to the work presented in this article. Rouse and Rouse [1980] studied displays for procedures in an abstract scenario. Three displays were used: a traditional hardcopy, a practically identical softcopy (displayed on a CRT screen), and a cued softcopy that dimmed a procedure step when it had been completed. To simulate the distractions faced by pilots, an arithmetic side task was added. The experimental results showed the cued softcopy display to be significantly faster and to cause fewer errors than the other two displays.

In a second, similar experiment conducted in a realistic environment, Rouse, Rouse, and Hammer [1982] studied hardcopy and cued procedure displays in a Link GAT-II twin engine aircraft simulator. Their experiment will be rather carefully described because the data from it was reanalyzed for the research reported here.

The aircraft simulator was configured as a Piper Aztec F. A PDP-11/40 minicomputer was interfaced to the simulator and recorded timestamped changes of the aircraft state. The record of a flight, termed a simulator trace, consisted of a sequen\e of triples, where each triple contained a time, a signal, and a signal value. A signal was recorded only at a significant change, which usually was a deviation of ±10% from its previously recorded value. (It is this data that was analyzed in the research reported here.) Also interfaced was a special purpose keyboard that controlled the CRT procedure display, one level of an independent, display variable. The other level was a traditional, printed hardcopy procedure.

Subjects flew in normal, emergency, and double emergency flights. The eight subjects were all instrument- and twin engine-rated pilots with the exception of one who had 70 hours of twin engine simulator time and was judged to have been equal to the others. Subjects flew in 3 flight scenarios. The _normal_ flight was a departure, climb to 2000 feet, direct cruise to another airport, descend, and land. The _emergency_ flight was a single engine failure after the aircraft climbed and through 2000 feet. The _double emergency_ failure consisted of a single engine failure at the same point plus a gear extension failure during the single engine pre-landing procedure.

The data analysis showed that the hardcopy procedure was 19% faster than the CRT display (statistically significant at $p<.025$). The CRT display produced 7.5 times fewer errors of the kind that could possibly be affected by display ($p<.025$).

A finer grained analysis of the experimental data from the above experiment appears in [Rouse and Rouse, 1983]. Forty-three errors were classified as occurring during hypothesis check, goal choice, procedure choice, or procedure execution. Errors were also classified as incorrect or unnecessary across all of these four categories. Displays were found to have a significant effect on errors that were categorized as wrong or incorrect. No effect was seen on errors classified as unnecessary actions.

PROGRAMMING METHODOLOGIES

The most appropriate programming methodology depends on the type of problem to be solved. This is true even though all methodologies are theoretically equal, since humans may find certain programs easier to express in one methodology than another. Though many methodologies exist, only two will be discussed here: conventional von Neumann programming [Backus 1978] and symbolic programming.

Conventional programming is often represented by the typical FORTRAN, BASIC, COBOL, PL/I, and Pascal program. Each computational step has one or more input values (or vectors) and produces a single output value (or vector). The values are usually numbers or characters. Such a methodology is best suited for numeric or data processing tasks such as aircraft simulator dynamics or implementing the lowest level of aiding: warning, calculation, and display control.

Symbolic computation, done primarily in Lisp or perhaps Prolog, is better suited to higher levels of aiding because the problem the human solves is itself symbolic. In other words, an aid should use symbolic computing to solve symbolic problems. The two methodologies employed are rule-based systems and scripts.

Rule-Based System

A rule-based system (RBS) [Waterman and Hayes-Roth, 1978] is one form of symbolic computation often used for an expert system, a program capable of rivaling human performance in a small but complex problem domain. Some examples of expert systems are:

1. MYCIN [Shortliffe, 1976] - selects antimicrobial therapy for infections.

2. DENDRAL [Buchanan et al., 1969] - analyzes mass spectroscopy data to reconstruct the original molecule from its constituents.

3. PONTIUS-0 [Goldstein and Grimson, 1977] is a system that achieves attitude instrument flight.

4. Wesson [1977] has produced a program to perform the enroute ATC function with performance (under real world conditions) as good as a human controller.

The structure of a RBS contains two principal parts: working memory and rules. For flight management, working memory can be assumed to contain the entire state of the aircraft (e.g. airspeed, altitude, pitch, roll, engine variables, electrical variables) as well as additional temporary memory. A rule contains two parts: a situation (such as altitude decreasing or airspeed > Vx) and an action (such as a procedure or storing some value in working memory). The following example shows possible rules for the pilot's handling engine failure during takeoff:

RULE	SITUATION	ACTION
1.	airspeed < Vmc	close throttles stop on runway
2.	Vmc < airspeed < Vx	abort flight close throttles stop on runway
3.	VMC < airspeed < Vx and sufficient runway	accelerate to Vx
4.	Vx < airspeed	maintain control and speed clean up aircraft climb secure engine land as soon as possible

The rules operate as follows. If the airspeed ≤ VMC when one engine fails (the aircraft is in contact with the runway), the situation of rule 1 applies, and the flight is aborted as per the action of rule 1. Rule 3 gives a further example of how rules are invoked. If Rule 3 is applied, airspeed will be increased to at least Vx, at which time Rule 4 applies. Thus, one rule may transfer control to another rule either by a change in the aircraft state as in this example or in temporary memory (not illustrated).

In the system discussed here, rules are used primarily for their ability to recognize situations. In other words, rules detect pilot actions and changes in the aircraft state (e.g., landing gear up) that indicate a new mode of operation (e.g., from on the ground to airborne). The rules, however, are not self-organized; they are held together by scripts.

<u>Scripts</u>

The script [Schank and Abelson 1977], the final programming methodology discussed here, is a form of symbolic computation like rule-based systems. Where a RBS recognizes specific situations and invokes the corresponding actions, a script describes the expected actors and actions in some situation. The script is a construct similar to the frame [Minsky, 1975] and to schema or template [Bartlett, 1932].

As an example of how the script concept might apply, consider a script for landing an airplane. The landing script provides the desired aircraft configuration -- engine settings, flaps, gear -- and their changes over time. Some of these will be dependent on the airport, and hence the landing script will have airport-dependent parameters. In addition, the landing script will indicate the scripts most likely to be activated next -- taxi, go-around, travel to an alternate airport, etc.

The power of scripts comes both from their rich description of actions and from the ability to determine which script is really active. The original application of scripts was understanding natural language (e.g., English). In spoken language, the speaker will, in the interest of economy, omit many details that the listener can infer. A script provides background for the computer so that it might draw some of the same inferences that a human would. To determine the next active script is a matter of selecting the script that best matches the current facts.

In a similar way, scripts can be used to infer what the flight crew is doing. The various controls and switch settings, sensed by the computer, can be viewed as a stream of details that partially conveys the flight crew's current thinking. By using scripts, the computer should be able to infer the full details in much the same way as it is used to understand natural language. In fact, one can envision an "advanced" intelligent cockpit where the computer would use the crew's conversation as one of its data sources. Though this may seem far-fetched, it will be demonstrated later that some errors could only be detected by this means.

PROCEDURES

Because the procedures pilots followed are central to this work, an example is given in Figure 1 of a typical procedure. Some aspects of these procedures will now be given. First, note that most steps are quite simple, e.g., steps 1 and 2; and the program senses their completion by a simple examination of the simulator state. Second, some steps cannot be sensed because the required signals are not available to the computer. An example is step 14; instrument vacuum was not recorded. Such steps are ignored by the program (deleted from the internal model at startup). Third, some steps call for the pilot to check a sensor reading. The program can check the sensor, but it cannot be sure the pilot has done so if the sensor reading is acceptable. Steps 10 through 13 are an example. Fourth, sensing some conditions may be difficult because the changes were not logged in the simulator trace because they were too small. For example, step 7 requires a 175 rpm drop, which is about 8% of the existing 2300 rpm. This change was unlikely to have been recorded in this data. Thus, the program can observe the magneto grounding but not its effect on engine rpm. This same problem also occurs when the pilot fine tunes the engines (leans mixture, changes propeller), as these changes are typically too small to be recorded. Of course, the problem of unavailable data would not be a problem in a real aircraft or in simulator data collection with a high sampling rate.

Some aspects of the simulator and aircraft in general make the sensing of steps more complex than it would first appear. First, some changes require time to occur. For example, in step 5, the propeller feather switch, which is discrete, may precede by a second the actual change of the propeller. A second difficulty is properly sensing temporary states. Two steps in the shutdown procedure, not shown, are an example. One is a momentary interruption of the magnetos and the other is a complete shutdown to stop the engines. Sensing the former requires that the transitions from ON to OFF to ON be observed within a short time frame. If this were not done, the program might misinterpret some other change to the magnetos.

INTERNAL PROGRAM STRUCTURE

The internal program structure will first be described in terms of a single procedure step. Next, the hierarchical organization of steps, procedures, and flight phases is described. Finally, the control structure, which interprets the steps, procedures, and phases, is described.

The first data structure is the aircraft database, which contains roughly seventy discrete and continuous signals. Each input record contains three items: a timestamp, a signal number, and a signal value*. As the input is read, new values are inserted in the database. Old values are not, however, immediately forgotten. Instead, they are retained if they are less than 6 minutes old or less than 100 in number so that the program may inspect earlier states.

The second data structure is the internal model of the procedures used by the crew. An individual procedure step (and other entities to be discussed later) is represented by the Pascal record shown in Figure 2. NAME is a text string that is used for humans to read. CAN_EXEC, DONE_EXEC, and ABORT_EXEC are rules (expressions that evaluate either true or false) that determine whether a step's STATE is considered UNSTARTED, IN_PROGRESS, DONE, or ABORTED according to the transition diagram shown in Figure 3. For example, for step 1 of the engine start procedure, the DONE_EXEC rule would check to see that both right and left mixture controls are currently at the full-rich setting.

ALLOWED et alia are sets of signals that can or cannot change during the execution of this step. These sets are used to detect actions that should not occur. When a simulator state change is read, these sets are examined to determine if the

---

*Other inputs -- keyboard entries, flight observer signals, etc. -- were ignored.

change is allowed. Only steps that are IN_PROGRESS are examined. Six sets were found necessary to detect pilot error. Normally, the program examines ALLOWED and DISALLOWED to determine the allowability of the signal. In engine-out emergency procedures, steps often refer to controls on the operative or inoperative engine. Thus, four more sets are necessary for the product of operative/inoperative with allowed/disallowed.

Procedure Step Hierarchy

Up to here, procedure steps have been described without mentioning their surrounding context. In fact, there is a hierarchy of four levels, with procedure steps at the lowest level. A number of steps are collected under a single procedure. One or more procedures are collected under a phase (e.g., pre-flight, takeoff), and all phases are collected under a single entity FLIGHT. Figure 4 illustrates the hierarchy. Each circle in Figure 4 corresponds to one script record as shown in Figure 2. Thus, all levels are represented uniformly. PARENT and COMP fields are used to represent the hierarchy.

The checkoff of procedures and phases is handled just as it is for procedure steps. There is some structure imposed on this process by the hierarchy, however. Only when a procedure STATE is IN_PROGRESS will its procedure steps (its subcomponents) be examined for transitions to new states. Further, when a procedure is DONE or ABORTED, its steps are not examined for transition. The structure imposes a preferred order of left-to-right on the execution of sub-scripts beneath a given script. The program expects execution in this order, but is capable of following any order. The program continually examines the DONE rules of all steps beneath an IN_PROGRESS procedure. The changing of the simulator state causes the rules to evaluate true in the order that the steps are completed.

The hierarchy also controls testing for allowed changes. First, only IN_PROGRESS steps, procedures, and phases are examined. All of the IN_PROGRESS steps are tested to determine if the signal is in one of the sets. If not, the same tests are made of IN_PROGRESS procedures, and, if necessary, of the IN_PROGRESS phases.

Emergency Daemon Procedures and Substitute Procedures

For the normal flight, the procedural hierarchy works well. During an emergency, flight operations are less structured. For example, a single engine-out emergency can happen any time the engines are running and the aircraft is airborne. Consequently, the procedure(s) for this situation must be available when the situation demands. Such procedures are termed daemons, and they were stored in a data structure separate from the normal procedures. The CAN_EXEC fields of these daemons look for the situations in which they are relevant.

A second modification for emergency procedures was substitute procedures. For example, in an engine-out emergency, the regular pre-landing procedure is replaced with a single engine pre-landing procedure. Substitute procedures were implemented by a pointer from the normal to the substitute.

EVALUATION AND RESULTS

The program was evaluated twice. The first time only normal procedures and normal flights were used. The program was then enhanced for the second evaluation, which used emergency and double emergency flights. The results for each evaluation are presented separately below.

Evaluation One

The program was first evaluated by developing the program on a derivation set of data and then running it unmodified on a validation set. The data was taken from normal flights and normal segments of emergency flights from the experiment of [Rouse, Rouse, and Hammer, 1982]. Flights were assigned randomly to derivation and validation data sets. As stated earlier, the objective was for the program to identify all of the errors found by Rouse and Rouse [1983].

The derivation data contained eight errors; as shown in Table 1 the program was able to locate seven of them positively and give an ambiguous indication of the eighth. This one error was omission of the cruise procedure and was originally located by examination of verbal transcripts. From the aircraft data recorded during the flight, the following can be determined. Of the three steps in the cruise procedure, the cowl flaps were definitely closed, the mixture might have been leaned (the necessary change might not have been enough for the computer to record), and the reduction in engine power was probably done, although one sensor reading required for the program to determine this was not available. It may be that the pilot executed the procedure without using its display or performing the callouts.

The validation data contained twelve errors. Eight errors were detected outright. Two errors were missed because a step was done incompletely and out-of-order. The program is designed to catch either of these errors individually; however, if both kinds of error are present in one step, the program will categorize it as done incompletely. Of the remaining two errors, one was turning a switch on, then off, then on, which was its intended state. The program simply checked off the step that required the switch to be on. At that time, the program did not test for conditions to be maintained. For the second evaluation, this shortcoming was fixed by the allowed field. The step that corresponds to, say, a switch change also ALLOWS that switch to change. When the step is checked off (i.e., DONE), its ALLOW field will no longer be checked. Since no other step will ALLOW the change, it would be detected if it occurs

The remaining error was an irrelevant action that would have been detected had it happened during an identified phase of the flight. Unfortunately, it happened between phases. Ideally, phases should overlap slightly so that the program has some phase to test the action. In the second evaluation, this shortcoming was fixed by having the ending of one phase force the next phase to begin.

The two types of error caught by the program are the following. One additional omitted procedure (besides the one mentioned earlier) was detected. Nine inappropriate actions were detected; most of these were activation of lights, etc. that were inappropriate at the times they occurred. Two instances were detected of lowering the landing gear at airspeeds higher than the maximum. Three instances were detected of not setting a control to the proper point. This included landing with partial instead of full flaps and not fully testing the ailerons before takeoff.

It might be expected that the program would find new errors that had been missed in the earlier investigation. It did not. While the program did turn up several cases of steps out-of-order, they were not really errors. For example, the step of retracting the flaps required so much time that the following step -- a discrete switch change -- could be completed while waiting for the flaps to retract. No new, substantial errors were found by the program.

Evaluation Two

The same methodology of derivation and validation data was used in the second evaluation. The results are shown in Table 2.

The one error the program did not detect was execution of two procedures when only one was needed (normal pre-landing and single engine pre-landing). The only detectable difference between these two is a single step -- the setting of the cowl flaps. At the time the procedure was invoked, the cowl flaps were in the position (one-half) that a step of the procedure requested they be. This step was immediately made DONE. Later, the pilot closed the flaps. The program, due to a simple bug in an ALLOW field, accepted this change. Eventually, the single engine pre-landing procedure was finished. The pilot then went through the normal pre-landing checklist, which resulted in no changes save for a different cowl flap setting. This change was detected as incorrect. If the simple bug were corrected, the program would not accept the first cowl flap change.

In addition to errors, the program identified several anomalies in pilot behavior. The most frequent was steps executed out-of-order. Expert opinion of these specific situations was that no error occurred. For example, the landing lights, navigation lights, and rotating beacon may be shut off in any order (once the propellers have stopped spinning) even though the procedure lists a specific order for them to be done.

These anomalies could be used for two kinds of improvements. The first would be to improve the program. In the above example, it would be better to express the ordering requirements semantically (e.g., engine off precedes beacon off) rather than by ordering. The second improvement could be to the procedures themselves. For example, flaps may not be extended above certain airspeeds. This restriction is not written in the Aztec procedures even though a similar restriction is written for landing gear, which is the step preceding flap extension. Such inadequacies could be found by coding the procedures in a program.

CONCLUSIONS

A model of procedure execution has been implemented in a computer program. It was tested on aircraft simulator data and was able to find practically all of the already known errors plus locate some new ones. While this serves as a practical test of the methodology, the implications of its aiding ability are more significant.

Using the model as an aid would have two benefits. The first, and most obvious, would be to detect and eliminate a great number of procedural errors. Perhaps surprisingly, this improvement comes with no additional pilot workload. A correctly functioning procedural aid would not need to communicate with the pilot except when an error was made.

The second benefit of the model would be display control. The latest generation aircraft are fitted with electronic displays that presumably could or do display procedures. The computer model of procedure execution could well be used to select and control displays, which might also result in an additional reduction in pilot workload.

ACKNOWLEDGMENT

The assistance of William B. Johnson, Sandra H. Rouse, and William B. Rouse is gratefully acknowledged.

This research was supported by the National Aeronautics and Space Administration under Ames Grant No. 2-123.

REFERENCES

Backus, J., "Can programming be liberated from the von Neumann style," Comm. ACM, Vol. 21, No. 8, 1978.

Bartlett, F. C., Remembering, Cambridge: London, 1932.

Buchanan, B. G., Sutherland, B., and Feigenbaum, E. A., "Heuristic Dendral: A program for generating explanatory hypotheses in organic chemistry," Machine Intelligence, Vol. 4, American Elsevier: New York, 1969.

Colley, R. W., "A transition control system for procedure prompting," Transactions of the American Nuclear Society, Vol 41, 1982.

Goldstein, I. P. and Grimson, E., "Annotated production systems: A model for skill acquisition," Proc. Fifth Inter. Joint Conf. Artificial Intelligence, 1977.

Goodstein, L. P., "Procedures for the operator -- Their role and support," IWG/NPPCI Specialists' Meeting on Procedure and Systems for Assisting an Operating during Normal and Anomalous Nuclear Power Plant Operation Situations, Garching, Federal Republic of Germany, December 1979.

Hammer, J. M. and Rouse, W. B., "Design of an intelligent computer-aided cockpit," Proceedings of the 1982 International Conference on Cybernetics and Society, pp. 449-453, Seattle, October 1982.

Minsky, M., "A framework for representing knowledge," in _The Psychology of Computer Vision_, P. H. Winston, ed., McGraw-Hill: New York, 1975.

Monan, W. P., "Distraction -- A human factor in air carrier hazard events," NASA Aviation Safety Reporting System, Ninth Quarterly Report, Moffett Field, CA: NASA-Ames Research Center, June 1979.

Norman, D. A., "Categorization of action slips," _Psychological Review_, Vol. 88, No. 1, 1981.

Rasmussen, J., "What can be learned from human error reports," in Proceedings of the NATO Conference on Changes in the Nature and Quality of Working Life, Thessalonica, Greece, 1979.

Rouse, S. H. and Rouse, W. B., "Computer-based manuals for procedural information," _IEEE Transactions on Systems, Man, and Cybernetics_, Vol. SMC-10, No. 8, August 1980.

Rouse, S. H., Rouse, W. B., and Hammer, J. M., "Design and evaluation of an onboard computer-based information system for aircraft," _IEEE Transactions on Systems, Man, and Cybernetics_, Vol. SMC-12, No. 4, July/August 1982.

Rouse, W. B. and Rouse, S. H., "Analysis and classification of human error," IEEE Transactions on Systems, Man, and Cybernetics, to appear, 1983.

Schank, R., and Abelson, R., Scripts, Plans, Goals, and Understanding, Lawrence Erlbaum: Hillsdale, NJ, 1977.

Seeman, S. E., Colley, R. W., and Stratton, R. C., "Optimization of the man-machine interface for lmfbrs," American Nuclear Society International Topical Meeting on LMFBR Safety and Related Design and Operational Aspects, Lyon, France, July 1982.

Shortliffe, E. H., Computer-Based Medical Consultations: MYCIN, American Elsevier: New York, 1976.

van Eekhout, J. M., and Rouse, W. B., "Human errors in detection, diagnosis, and compensation for failures in the engine control room of a supertanker," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 12, December 1981.

Waterman, D. A., and Hayes-Roth, F., Pattern-Directed Inference Systems, Academic Press: New York, 1978.

Wesson, R. B., "Planning in the world of the air traffic controller," Proc. Fifth Inter. Joint Conf. Artificial Intelligence, Cambridge, MA, 1977.

Wiener, E. L. and Curry, R. E., "Flight-deck automation: promises and problems," Ergonomics, Vol. 23, No. 10, 1980.

1. Mixture controls	full rich
2. Propellers	full high rpm
3. Throttles	set 2300 rpm
4. Propellers	exercise 300 rpm max drop
5. Propellers	feather check 500 rpm max drop
6. Magnetos	check
7.	175 rpm max drop
8.	50 rpm max differential
9. Engine gauges	check
10.	oil pressure
11.	oil temperature
12.	cylinder head temperature
13.	ammeter
14.	vacuum
15. Throttles	set 1000 rpm

Figure 1. Engine Run-up Procedure

```
SCRIPT               = record
    NAME             : string;
    CAN_EXEC         : rule;
    DONE_EXEC        : rule;
    ABORT_EXEC       : rule;
    STATE            : [UNSTARTED,IN_PROGRESS,DONE,ABORTED];
    ALLOWED,
    DISALLOWED,
    OP_ALLOWED,
    INOP_ALLOWED,
    OP_DISALLOWED,
    INOP_DISALLOWED   : set of signal;
    COMP              : array[1..30] of Uscript;
    PARENT            : Uscript;
    SUB               : Uscript;
    end;
```

Figure 2.   Script fields.

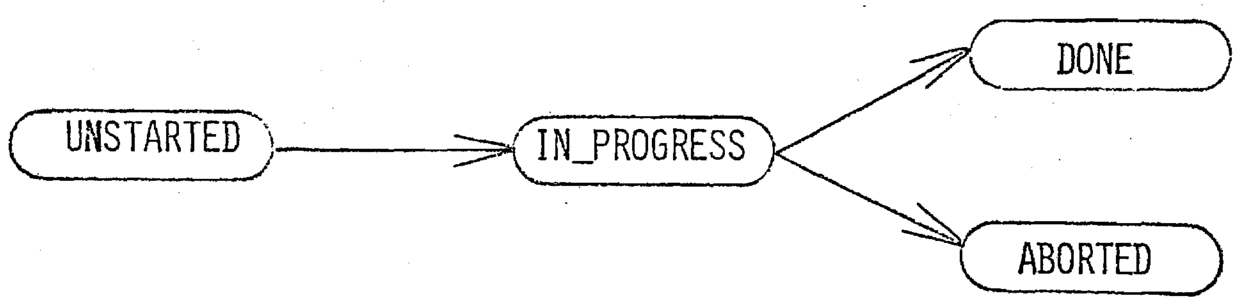


Figure 3. STATE transitions

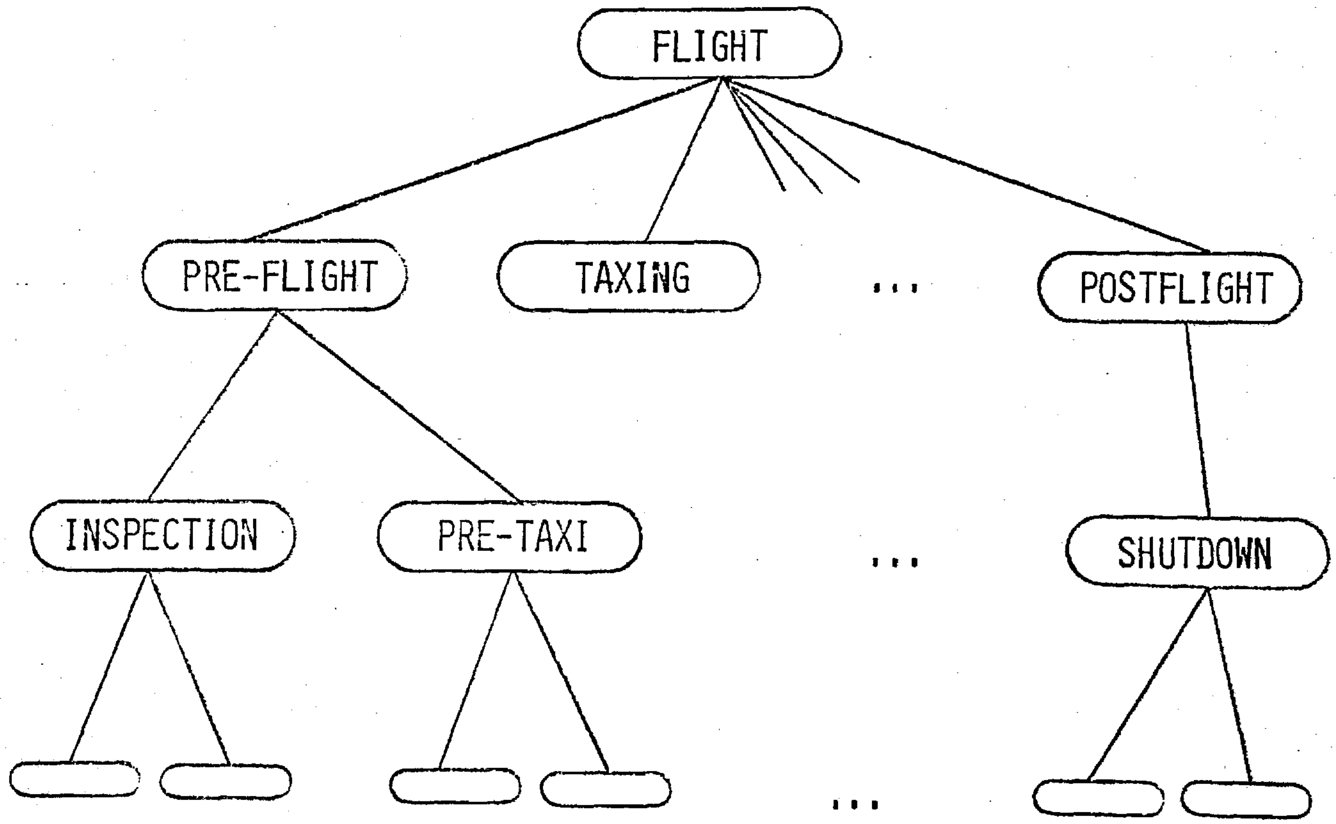


Figure 4. Hierarchy of steps, procedures, phases, and FLIGHT.

	Refound	Missed	Undetectable	New
Derivation	7	0	1	0
Validation	8	4	0	0

Table 1. Normal flight error analysis.

Refound errors were found by Rouse, Rouse, and Hammer and by the program. Missed errors were found by the original investigators but not by the program. Undetectable errors were found by the original investigators using source of data (i.e., cockpit tape recordings) that were not available to the program. New errors were found by the program but not by the original investigators.

	Refound	Missed	Undetectable	New
Derivation	6	0	4	3
Validation	9	1	4	5

Table 2. Emergency flight error analysis.

THE EFFECTS OF TYPE OF KNOWLEDGE UPON HUMAN PROBLEM SOLVING IN A PROCESS CONTROL TASK

Nancy M. Morris and William B. Rouse

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The question of what the operator of a dynamic system needs to know was investigated in an experiment using PLANT, a generic simulation of a dynamic production process. Knowledge of PLANT was manipulated via different types of instruction, so that four different groups were created: 1) Minimal instructions only; 2) Minimal instructions + guidelines for operation (Procedures); 3) Minimal instructions + dynamic relationships (Principles); 4) Minimal instructions + Procedures + Principles. Subjects controlled PLANT in a variety of situations which required maintaining production while also diagnosing familiar and unfamiliar failures. Despite the fact that these manipulations resulted in differences in subjects' knowledge as assessed via a written test at the end of the experiment, instructions had no effect upon achievement of the primary goal of production, or upon subjects' ability to diagnose unfamiliar failures. However, those groups receiving Procedures controlled the system in a more stable manner. Possible reasons for the failure to find an effect of Principles are presented, and the implications of these results for operator training and aiding are discussed.

INTRODUCTION

The role of operators of engineering systems, such as aircraft, ships, or process plants, has changed greatly in recent years and continues to change. Much of this change has been precipitated by advances in automatic control of systems. As the responsibility for control is shifted to computers, the operator becomes less a controller and more a monitor and, if necessary, a problem solver [1]. As a result, the operator of an automatically controlled system is called upon to exercise quite different skills from those required of the operator of a manually controlled system. Beyond some minimal level, psychomotor ability becomes less essential, and greater emphasis is placed upon the use of cognitive skills such as reasoning, pattern matching, and problem solving.

Realizing this, a variety of individuals concerned with system design and operator training have argued that one should "consider the cognitive processes of the operator" when dealing with design and training issues (e.g., [2], [3]]). Few people would dispute this idea, because the assertion that the operator's needs and capabilities should be considered seems to be a reasonable one. However, further development of the concept as stated here is required if it is to be practically useful.

From a theoretical viewpoint, theorists and researchers in the fields of psychology and artificial intelligence as well as within the domain of process control have discussed human cognition in a variety of problem situations. A number of models of reasoning and decision making have been offered, employing such concepts as schemas, scripts, heuristics, etc. (e.g., [4], [5], [6]). The general opinion is that, when faced with a problem, the human uses some understanding mechanisms which govern the situation in making decisions.

A construct which appears in many writings is that of the "mental model" of the process (e.g., [7], [8], [9], [10], [11], [12]). Although, unfortunately, the term has sometimes been employed rather loosely, the mental model has generally been used as a representation of knowledge of a system and its relationship with the environment. A number of functions have been attributed to the mental model, including guiding information seeking [11], [13], [14], aiding in pattern recognition [15], [16], and anticipating future system states [17].

One of the most articulate discussions of the relationships between mental models and problem solving in operation of engineering systems has been provided by Rasmussen [18]. In ordinary, familiar circumstances, the human operator appears to rely upon available heuristics and rules of operation. In other words, the operator's behavior is rule-based. However, in unusual situations for which rules do not apply, the human operator must reason at a knowledge-based level, using an understanding of the functioning of the system to determine an appropriate course of action. Thus, different mental models may be more or less appropriate, depending upon the situation.

From a practical perspective, the idea of considering operator cognitive processes and the notion that multi-level reasoning may be required have attracted the interest of those concerned with system design and the training of operators [19]. Practitioners have found, however, that the manner in which system designs and training programs should be structured so as to incorporate these ideas is not at all straightforward. For example, it has been suggested that one should strive to support the operator's reasoning and decision making process by providing information that enhances the operator's model [16], [20]. Yet, translating this suggestion into a specific course of action is not an easy task.

Speculation has been directed at the nature of the mental model associated with good performance. As a result of this speculation, it has been assumed both implicitly and explicitly that an important part of the mental model is a representation of the dynamics of the system. Some educators have further stated that such a representation (i.e., a "thorough understanding of the dynamics of the system") is a requisite if the operator is to be effective (e.g., [21]). Based upon this assumption, training programs may be aimed at providing the operator with the appropriate mental model, usually via instruction in the theory upon which the system is based and perhaps some experience with simulators. Often the further assumption that such instruction will lead to satisfactory performance is made.

Unfortunately, although these approaches may be intuitively appealing, there appears to be little in the way of empirical support to guide the practitioner's efforts. For example, there is lttle or no conclusive evidence that providing operators with information about theoretical aspects of system functioning enables them to be better operators. In fact, in research in which subjects were given instruction in the theoretical basis of system functioning there was no apparent advantage to having been given such information [9], [22], [23], [24]. It is quite possible that being able to control the system is not directly related to an explicit knowledge of system dynamics. Alternatively, it is conceivable that effective control behavior may be related to having an understanding of system dynamics, but that this understanding may be in the form of a "process feel" and may not be obtained via verbal instruction. At any rate, in spite of the lack of support for the practice, there is continued emphasis on instructing operators in the theoretical basis of system functioning.

The experiment reported in this paper was designed to investigate the question of what the operator of a dynamic system needs to know in order to be effective. In particular, the value of two different types of knowledge--knowledge of how to control the system, and knowledge of how the system works--was explored. The general approach was to manipulate system-relevant knowledge via instructions, and examine the effects of this knowledge upon performance.

A PROCESS CONTROL TASK

This research was conducted in the context of PLANT, a computer-driven generic simulation of a dynamic production process. A graphic display for a sample PLANT problem is shown in Figure 1, and the information display which accompanies the graphic display is shown in Figure 2. A general description of PLANT is presented here. Interested readers are referred to [25] for further details about the simulation.

Referring to Figure 1, in this system there are nine tanks, some of which are currently connected by open valves (represented by lines between tanks). Fluid enters the PLANT system at the left and exits at the right as finished product. In general, the PLANT operator's task is to supervise the flow of fluid through the series of tanks interconnected by pumps, valves, and pipes so as to produce an unspecified product. The operator may open and close valves, adjust system input and output, check flows between tanks, and order repairs of various PLANT components, in order to achieve the primary goal of maximizing production.

Each operator action, such as opening a valve or adjusting input, requires one time unit or iteration. PLANT is not updated automatically in real time, but rather is at steady-state between commands and is thus self-paced. Although it is possible for PLANT to run in a forced-paced mode and periodically update automatically (e.g., once every four seconds), the decision was made to employ the self-paced mode of updating because of the long response times characteristic of real processes.

As in real systems, although maximizing production is the primary goal of PLANT operation. the "physical" limitations of the system (such as tank or valve capacity or reliability of system components) require that the PLANT operator be concerned with secondary goals as well. Among these secondary goals are stabilization of the system, and detection, diagnosis, and compensation for system failures. Stability is required because of the dynamic characteristics of the system* and the fact that PLANT valves do not have infinite capacity. Should the operator fail to maintain stability, the PLANT safety system intervenes in order to protect the system from damage due to unsafe operating practices. The safety system operates by automatically closing valves (i.e., "tripping" them) and/or stopping system input or output if flows or fluid levels exceed desired ranges.

Possible PLANT failures include valve failures, pump failures, tank ruptures, and failure of the safety system. Valve and pump failures are fairly common, and involve a stoppage of flow between connected tanks. While flow is stopped, the display remains unchanged and, therefore, the failed valve or pump appears to be working. Detection and diagnosis of a valve or pump failure may be accomplished by noting a difference in observed and expected fluid levels in tanks, and checking flows through the suspected valve(s). Repair involves sending a "repair crew" to the site of the failure for a period of 5-10

*Each pair of connected tanks is modeled as a second-order system with rate of flow and its derivative as state variables and transition matrix determined by pipe and tank cross-sectional areas, pipe lengths. and fluid characteristics. See [26] for a derivation of the state equations.

iterations.

Tank ruptures and failure of the safety system are extremely rare by design, and may occur only once during a subject's experience with the system. As a result, these failures provide a means for studying operator problem solving in unfamiliar situations. A tank rupture must be inferred from noting a loss of resources from the system, and occupies the repair crew for 15 iterations, during which the tank is drained and "patched".

The nature of the failure of the safety system failure is much less predictable due to the range of possible safety system actions; it may be manifest by a number of different symptoms, and may be intermittent. For example. failure of the safety system could result in arbitrary "tripping" that should not be difficult to detect if one understands the way in which the safety system works. Thus, detection and diagnosis of a safety system failure requires that the operator have some knowledge of the functioning of the safety system and the underlying dynamics of the process, because safety system actions are directly related to PLANT dynamics. During repair, the safety system is deactivated for 20 iterations and the operator is responsible for PLANT safety.

With respect to the PLANT environment, it is possible to identify different types of knowledge about PLANT which the operator might have. At a minimum, he might know that he is controlling a process, his goal is to maximize production, and that various control options are available. At another level, the PLANT operator may know "what to do" in certain situations--i.e., he may have a set of procedures or rules which, when followed, enable adequate control of the system. Finally, it is possible for the operator to have a knowledge of the way in which PLANT "works", including an understanding of the underlying process dynamics and relationships between components.

In the research described in the following section, an attempt was made to "create" operators with these different types of knowledge by providing naive subjects with differing instructions. These operators were then placed in familiar and unfamiliar situations in order to provide them opportunities to use the information they were given. During the planning and conduct of this research, the following outcomes were expected. First, it was anticipated that those operators with a set of procedures for controlling PLANT would be better in ordinary or familiar situations than those without such information. Second, it was predicted that those persons with an understanding of PLANT dynamics and principles would be better able to deal with unfamiliar situations.

METHOD

Subjects

Junior and senior undergraduates at Georgia Institute of Technology served as paid volunteer subjects. All 32 were industrial and systems engineering majors, and had completed courses in physics, dynamics, and higher level mathematics.

It is important to note here that, although the use of students as subjects is often considered to compromise credibility in applied research. this subject population was well-suited to the questions at hand. This is due to the fact that operators in many systems (e.g., nuclear power plants) are required to complete a training program which is technically equivalent to that required for a bachelor's degree in engineering. Therefore, it is argued that these students had educational backgrounds comparable to actual operator trainees in some domains.

Experimental Materials

Four sets of written instructions relevant to PLANT were used in the experiment: Minimal instructions, Principles, Procedures, and Relationships Between Principles and Procedures. The format for the first three was similar, in that each consisted of text interspersed with "self-test" questions and accompanied by 1-2 page summaries of important concepts. The fourth set, Relationships, differed, as it was designed to be inserted into Procedures for an experimental group which was instructed using both Procedures and Principles. These instructions were designed to represent the types of knowledge about PLANT discussed earlier. (The complete sets of instructional materials appear in Morris' thesis [1].)

Minimal instructions were directed at what questions: what kind of system is it, what is the goal of operation, what can happen, etc. As such, Minimal instructions consisted of an introduction to the concept of a process plant, and a discussion of the goals of PLANT operation, operational constraints, possible malfunctions, and command options available. Self-test questions in the Minimal instructions were directed at insuring an understanding of the basics of PLANT operation (such as opening valves and adjusting input and output) and the nature of the PLANT safety system and possible PLANT malfunctions.

Procedures told the PLANT operator how the system should be controlled, in both general and more specific terms. First, there were three heuristics useful for general control of PLANT (e.g., "keep all valves open"). The Procedures also included a set of six more specific sequences of control actions (i.e., procedures in the formal sense) appropriate for use in a number of undesirable PLANT states (e.g., "output column too low"). These "specific sequences" were not as specific as the procedures used in aviation, but were more like "guidelines", discussing appropriate types of control actions rather than specific commands to be entered. The majority of the self-test questions required the subjects to determine which procedure was applicable in a depicted PLANT state (i.e., "Which procedure would you choose in this situation?").

These procedures were the product of numerous discussions between the authors. each of whom had considerable experience in controlling PLANT and had developed his/her own strategy for doing so. Procedures were evaluated for their "reasonableness" by actually using them to control the process; in instances where alternative procedures had been generated, the sequence of steps leading to the best performance (i.e., the most production and fewest valve trips) was selected.*

Principles included a presentation of an approximation of the state equations governing PLANT dynamics, and a verbal interpretation of the equations in terms of observable dynamic relationships. In short, the Principles indirectly contained information as to _why_ PLANT should be controlled in a certain manner. In writing the Principles, an effort was made to make them as meaningful and relevant to PLANT operation as possible. Discussion of abstract theory was avoided, and mathematical expressions were always limited to simple algebraic expressions and accompanied by a discussion of their meaning and importance to PLANT functioning. For example. the instructions stated that the PLANT was "sluggish", that flows tended to "oscillate over time", and that input into a tank was "shared" by the valves leading from it. Self-test questions required the subject to apply the written information to the solution of problems (e.g., "If tank B had a level of 75 and tank F had a level of 63 when valve BF was opened, what would be your estimate of the initial flow rate for valve BF?").

Relationships Between Principles and Procedures were more directly related to the "whys" of PLANT operation. In Relationships, the rationale behind the information in the

---

*Throughout this paper, reference is made both to the set of procedural instructions and to operational procedures found in these instructions. To avoid confusion. references to the instruction set begin with an upper-case letter (i.e., Procedures), whereas "procedures" refers to specific sequences of steps found in the procedural instructions.

Procedures was presented in terms of concepts discussed in the Principles. Generally, subjects were informed, "You should (do this) because (the PLANT works this way)". As noted earlier, Relationships was inserted in Procedures for an experimental group which was instructed using both Procedures and Principles.

Two multiple-choice tests of the information in the instructions were also used. Test 1 contained 22 items, all related to information in the Minimal instructions. Test 2 consisted of 54 items, with approximately one third devoted to each of the major types of instruction (i.e., Minimal, Principles, and Procedures). Minimal questions on Test 2 were virtually identical to those on Test 1, with minor modifications. When creating procedural and principle questions, an effort was made to avoid asking questions which would be impossible to answer correctly without having been explicitly told the answer in instructions. For example, alternative answers often consisted of a range of numbers rather than specific values.

Experimental Method

Subjects served in a total of 12 sessions each, with the average length of each session being approximately 60 to 75 minutes. With the exception of sessions 10 and 12 (which were counterbalanced), the order of presentation of PLANT production runs was identical for all subjects. The first eight sessions were training sessions, in which subjects received written and oral instructions and controlled PLANT in a variety of situations for varying lengths of time. Material presented in instructions was repeatedly reviewed during training sessions, and all of subjects' questions were answered, if possible. in a manner appropriate to a particular subject's experimental condition.

The last four sessions were experimental sessions, and were identical in terms of initial PLANT configuration and length of production run. Sessions 9 and 11 were "familiar" runs, in that all failures which occurred were failures which the subjects had experienced before (i.e., valve and pump failures). Sessions 10 and 12 were "unfamiliar" runs, each involving a malfunction which had been discussed in instructional materials but which had never occurred in a subject's experience (i.e., tank rupture and safety system failure). The type of unfamiliar failure which occurred was counterbalanced across subjects and within instructional groups (described later). No instructions from the experimenter were provided during the last four sessions, and no questions from subjects were answered.

All subjects were presented with the Minimal instructions at the beginning of session 1, and were allowed to read them with the understanding that they would always have access to written materials when controlling PLANT. Following an oral review of the instructions with the experimenter, they were allowed to control PLANT for approximately one hour. During their first production run, they were encouraged to try all commands to make sure they understood how they worked.

Session 2 consisted of a brief review of commands and another one-hour production run. Test 1 was administered at the end of session 2. Since it was intended primarily as a vehicle for discussion, all correct and incorrect answers were discussed with subjects and important points were emphasized. Sessions 3 through 7 were "problem" runs, with subjects assuming control of the PLANT in a variety of unstable situations. These problems were created by the experimenter, and represented situations for which specific procedures were applicable. Sessions 8 through 12 were "normal" runs once more; as in sessions 1 and 2, no problems existed when the subject began controlling the PLANT. Test 2 was administered at the end of session 12.

Differentiation of experimental groups began in session 3. At the beginning of session 3, two groups of eight subjects each (groups B and D) were given Principles, and a third group (group C) was given Procedures. The remaining eight subjects (group A) were given no further written instructions. At the beginning of session 5, subjects in group D were also given Procedures, with Relationships Between Principles and Procedures inserted at the appropriate point.

To summarize, group A received Minimal instructions; group B received Minimal instructions and Principles; group C received Minimal instructions and Procedures; group D received all instructions. These four groups may be viewed as cells in a 2 x 2 factorial design, with each group receiving Procedures or no Procedures, and Principles or no Principles.

A number of measures of subjects' performance were recorded. In addition to the obvious performance measure of production, several intermediate measures were noted as indications of how "elegantly" subjects achieved their goal. Among these were the number of automatic valve trips, number of limit alarms (i.e., tank levels too high or too low), number of valves open per iteration, number of observations made prior to repairing a failure, variability of fluid levels both within and between columns, and frequencies of various commands.

RESULTS

Analysis of variance was used as the primary statistical tool for data analysis. Performance measures were used as dependent variables in three-way analyses with two between-groups factors (Principles and Procedures) and one within-groups factor or repeated measure (session). The following results are presented to provide an overview of the experimental findings. A more in-depth analysis of the results of this research may be found in [1].

When production achieved was used as the dependent variable in the analysis, there was no effect of either Procedures or Principles. The interaction also failed to reach significance. Of all the other performance measures, there were three which revealed significant differences related to instructions. These were the average number of automatic valve trips, average number of valves open at any point in time, and variance of fluid levels (i.e., tank heights) within the system.

All of the significant effects upon these variables were those of Procedures. Subjects provided Procedures (i.e., groups C and D) generally experienced fewer automatic valve trips (.94 vs. .66 per iteration, p = .0343), kept more valves open (15.79 vs. 14.58, p = .0074), and had less variance in tank heights (15.92 vs. 21.59, p = .0251) than did those subjects who did not receive Procedures (i.e., groups A and B). None of the main effects of Principles nor any of the Principles x Procedures interactions reached significance.

With regard to the unfamiliar failures, there was no difference in groups' ability to detect and repair the tank rupture or safety system failure. Only one person (from group D) did not repair the tank rupture, and approximately half in each instruction group repaired the safety system. Subjects were classified according to whether or not they repaired the failure of the safety system and the analysis of variance was repeated. (This classification is denoted by "fix-nofix" in the following discussion.) When differences in the variables noted above were analyzed in this manner, the following significant effects were noted.

First, those subjects who were able to determine that the safety system had failed generally produced more, regardless of session, than did those who did not make an appropriate diagnosis (321.3 vs. 298.7 units per iteration, p = .0303). Furthermore, "fixers" generally had fewer automatic valve trips (.68 vs. .94 per iteration. p = .0100), more valves open (15.64 vs. 14.68, p = .0128), and less variance in tank heights (15.92 vs. 21.59,

p = .0001).

With respect to two of these variables, trips and height variance, the interaction of Procedures and fix-nofix was also significant (p = .0031 and p = .0061, respectively). Analysis of the simple main effects of these interactions revealed that the differences were among those subjects who did not repair the safety system. In other words, those persons who repaired the safety system were equivalent in terms of trips and height variance, regardless of whether or not they had been given Procedures. Among those persons who did not repair the safety system, however, those people who were not given Procedures had more valve trips (1.30 vs. 0.65) and height variance (28.32 vs. 16.15) than those who received Procedures.

Differences in performance on Test 2 were also identified via analysis of variance. When overall scores were compared, there were significant main effects both of Procedures and Principles (p = .0008 and p = .001, respectively). Groups receiving Procedures scored higher than those receiving no Procedures (80.44% vs. 70.94%), and Principles groups scored higher than those not receiving Principles (80.09% vs. 71.30%). The interaction of Procedures and Principles was not statistically significant.

Comparing scores on test sections (i.e., questions related to Minimal instructions, Procedures, and Principles), the interactions of Procedures x section (p = .0128) and Principles x section (p = .0008) were significant. Analysis of simple main effects revealed that subjects receiving Procedures answered more procedural questions correctly than those who did not receive Procedures (82.53% vs. 61.33%), and subjects given Principles correctly answered more questions related to system dynamics (72.13% vs. 48.07%).

Correlations between all dependent measures were computed, and a subset of these correlations may be found in Table 1. The results of the analyses presented earlier clearly demonstrate the existence of some strong relationships between variables. These correlations are offered as a mechanism for integrating the more detailed results into an overall picture, which is discussed in the following section.

DISCUSSION AND CONCLUSIONS

Interpretation of Results

There are three observations which may be made relative to the information presented in Table 1. First, the significant correlations between production, trips, number of open valves, and variance in tank heights are noteworthy because they provide support for the information found in Procedures. The main thrusts of these guidelines were aimed at keeping all valves open and controlling differences in tank heights. Judging from the relationships of height variance, etc. to overall production, these emphases were well-founded. The point is necessarily made because it is unreasonable to expect operators to follow rules which are not appropriate.

Second, the high correlations between number of valve trips, number of valves open. and variance in tank heights reflect characteristics of PLANT and provide justification for the treatment of these variables as alternative measures of a single construct. Thus, a "stable" PLANT is one in which most valves are open, there are few valve trips, and there is little variance in tank heights. The concept of PLANT stability is utilized in the following paragraphs when differences in control performance are discussed.

Third, perhaps the most important observation to be made from an examination of Table 1 is that the relationship between PLANT performance and Test 2 performance was not very strong. The highest correlation between PLANT production and any measure of Test 2 performance was .19, which was not significant. Of all the correlations between Test 2 and PLANT performance. only the relationship between number of open valves and score on the procedural section of Test 2 achieved significance.

Between group variations on Test 2 indicate that manipulation of instructions relative to PLANT was at least moderately successful in establishing different groups with respect to PLANT-relevant knowledge. In fact, the pattern of test results obtained is exactly as one might predict would occur if the manipulation were successful. It is also interesting to note that, since the interaction of Principles and Procedures was not significant, the effect of providing more than one set of instructions was approximately additive.

In contrast to the results on Test 2, instructions were not as clearly reflected in PLANT performance. For example, instructions had no effect upon how much subjects were able to produce. Regardless of instructions, groups were able to achieve comparable production scores. Although production was comparable across groups, those groups receiving Procedures (groups C and D) controlled PLANT in a more stable manner than did the groups without Procedures (groups A and B). The provision of Principles did not seem to improve subjects' control behavior under normal circumstances.

Variations in instructions had no effect upon whether or not a subject was able to correctly diagnose the unfamiliar failure of the safety system. Judging from the analysis of the Procedures x fix-nofix interaction, a stable system was apparently a necessary prerequisite to finding this malfunction. This is not surprising, since there would be a greater contrast between "normal" and "abnormal" in such a system. However, it is also apparent that having a stable system was not a sufficient condition for the location of the safety system failure. Procedures enabled subjects to have a more stable system, but only half of those subjects receiving Procedures repaired the safety system.

<u>Restatement of Experimental Hypotheses</u>

Now, consider the results of this experiment in light of the experimental hypotheses stated earlier. To reiterate, the first hypothesis was that those groups receiving Procedures (i.e., groups C and D) would be better at controlling PLANT in ordinary circumstances than those not provided Procedures (i.e., groups A and B). The data obtained in this research support this hypothesis. Although there were no differences between groups in overall production achieved, subjects in groups C and D generally controlled PLANT in a more stable manner, and were more consistent with each other with respect to most dependent measures. This evidence indicates (to no great surprise) that proceduralization may indeed be a means of providing operators with an effective strategy, and thus supports the common practice of providing operators with procedures.

The second hypothesis was that persons with an understanding of the dynamics of PLANT as described in Principles (i.e., groups B and D, or at least group D) would perform better in unusual circumstances in which available procedures were not applicable. The results reported here provide absolutely no support for this hypothesis. As reported earlier, only one person failed to repair the unfamiliar tank rupture, and approximately half of the subjects in <u>each</u> instruction group repaired the failed safety system. In retrospect, all subjects had been told in the Minimal instructions how to detect a tank rupture, so this failure to note a difference between groups in repair of the tank rupture is not too surprising; however, the pattern of results obtained with the safety system failure was not expected, and is difficult to explain.

The provision of Principles did not insure that subjects would be able to deal with the unfamiliar safety system failure. Neither did Principles appear to be useful in ordinary situations, as group B was no better than group A in controlling PLANT. In light of the performance on Test 2, it may be stated that this does not reflect a failure on the part of subjects to learn the material. Nor does it appear that this failure to find an effect may be attributed merely to failure to achieve the traditionally accepted significance level of .05. In all cases, measured differences due to an effect of Principles were small, and the probabilities of these differences being due to chance were quite large.

Why not Principles?

There are two questions which immediately come to mind when considering the failure to find support for the second hypothesis. The first is this: Why did Principles fail to help? It is necessary to address this question because of prevailing opinion as to the value of such a knowledge--the Principles should have helped. In fact, this attitude is so firmly held that some may even be led to discount the results reported here, because "everyone knows that you need to understand how a system works in order to control it".

In considering this question of why the provision of Principles did not lead to better performance, it is important to note that these results do not appear to represent an isolated case. Rather, they are in agreement with the results of other research in which knowledge of theory was found to have little or no relationship to task performance [9], [22], [23], [24], [27]. In fact, a survey of relevant literature failed to reveal any reports in which a statistically significant advantage of such knowledge was reported, although many authors stated or implied that there was such an advantage.

One approach to explaining these results might be to argue that the effects of knowledge of theoretical principles may be indirect and subtle, and thus not directly measurable. Indeed, a number of more subtle effects seem feasible, though a detailed examination of this data fails to support them. For example, a general understanding of the functioning of a system may serve as a frame of reference from which procedures may be more meaningful and better understood. Understanding how the system works may have a motivational effect upon operators. Although such knowledge may not be useful to a group of operators as a whole, some individuals may find this information extremely useful.

An additional explanation for this consistent failure to find an advandage of theoretical instruction may be in terms of different types of knowledge. The results of this research suggest that knowledge of a system may be represented in more than one form, and that any given person's knowledge may consist of multiple representations. Thus, knowledge of "facts", as measured by a verbal test, and knowledge of how to control a system, as manifest by adequate control performance, may not be strongly related and may be embodied in different forms and thus expressed in different ways. The low correlations between Test 2 scores and PLANT performance measures are consistent with this interpretation.

If this is the case, then the impact of Principles may have been minimal because the information was not in a form that was directly usable by subjects (i.e., was not directly related to what they should be able to _do_. as opposed to what they should _know_). Rather, in order to apply the information appropriately, the operator first had to go through a deductive process. Either people did not attempt to do so, or did try but could not determine an appropriate course of action. In the absence of successful reasoning, the Principles could not be useful.

Alternatives to Principles

The second question which arises when considering these results is this: If telling operators how the system works does not insure that they will be able to deal with unanticipated events, then what can be done to provide such assurance? This reflects a pressing need in industry, because it is precisely for the purpose of handling unforeseen situations that human operators are employed. Accordingly, an attempt will be made to address this issue here.

It is appropriate to recall the concept of multiple levels of reasoning discussed earlier. People commonly engage in rule-based behavior when controlling familiar systems under normal conditions, but should resort to knowledge-based behavior in unusual circumstances, using an understanding of the way the system works to determine what should be done. Therefore, if a person has a knowledge base sufficient to support knowledge-based reasoning, this information should be used in unfamiliar situations. Although this seems to be a reasonable description of what should occur, the indications from this research are that this describes the ideal and not what actually takes place. As we have seen, knowledge and opportunity do not guarantee that people will engage in knowledge-based reasoning and reach an appropriate conclusion.

It seems that certain conditions must be met for a person to solve an unfamiliar problem successfully. First, he or she must have an adequate knowledge base. Second, it must be apparent that available rules do not apply and that reasoning about the problem is required. Third, the person must be able to use the information in the knowledge base appropriately to reach a conclusion.

The nature of this "adequate" knowledge base was the primary question pursued in this research. and the partial answer obtained was "less than one might suppose". Some subjects from groups A and C found the safety system failure, and were generally quite good at controlling PLANT, yet could not answer questions on Test 2 about PLANT functioning. While it cannot be stated that these persons had no ideas of how PLANT works, it can be said that their knowledge of PLANT was at least less detailed than the information contained in Principles. Therefore, it appears that the importance of a detailed theoretical knowledge of a system to an operator's control behavior has been overemphasized in training, and this emphasis should be reduced.

Therefore, it may be necessary to provide the operator some assistance at the time of the unanticipated event, possibly online. One form of assistance could be to adequately inform the operator that an unusual condition existed. Other authors have indicated that it might also be necessary to help him to pinpoint the location of the problem. Finally, it could be necessary to guide the operator in his reasoning process, to increase the likelihood that an appropriate conclusion will be reached. Research in the areas of decision making and decision aiding is moving in the direction espoused in this paragraph [28]. However, any existing operator support systems of the type envisioned here are mainly in the conceptual stage and little evaluative data is available.

Summary

In summary, the question of what an operator needs to know is extremely important to those responsible for operator training. Traditionally, operators have been required to learn a great deal about the theoretical aspects of system functioning, in the hopes of insuring that they can deal with unanticipated events. Available research evidence suggest that this emphasis on the importance of theoretical knowledge of the system is disproportionate to the actual value of such knowledge, and that more attention should be devoted to providing operators assistance during abnormal conditions. In other words, less emphasis should be placed on answering the question of "What does the operator need to know?" and more on the questions of "What should operators be able to do?" and "How can we help them to use the knowledge they have?"

ACKNOWLEDGEMENT

This research was supported by the National Aeronautics and Space Administration under Ames Grant No. NAG 2-123.

REFERENCES

[1] N. M. Morris, "Human problem solving in process control." Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, Report No. 83-2, 1983.

[2] D. A. Norman, "Human cognition and skills." Proceedings of the Workshop on Cognitive Modeling of Nuclear Plant Control Room Operators, pp. 92-106, 1982.

[3] T. B. Sheridan, "Cognitive models and computer aids for nuclear plant control room operators," Proceedings of the Workshop on Cognitive Modeling of Nuclear Plant Control Room Operators, pp. 52-62, 1982.

[4] W. B. Rouse and R. M. Hunt, "Human problem solving performance in fault diagnosis tasks." In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. 1). Greenwich, Connecticut: JAI Press, 1983.

[5] D. E. Rumelhart and A. A. Abrahamson, "A model of analogical reasoning," Cognitive Psychology, vol. 5, pp. 1-28, 1973.

[6] R. C. Schank and R. P. Abelson, Scripts, Plans, Goals, and Understanding. Hillsdale. N. J.: Lawrence Erlbaum, 1977.

[7] S. N. Anyakora and F. P. Lees. "Detection of instrument malfunction by the process operator," Chemical Engineer, vol. 264, p. 304, 1972.

[8] C. L. Borgman, "Mental models: ways of looking at a system," ASIS Bulletin, vol. 9, pp. 38-39, 1982.

[9] H. Kragt and J. A. Landeweerd, "Mental skills and process control." In E. Edwards and F. P. Lees (Eds.), The Human Operator in Process Control. London: Taylor and Francis, 1974.

[10] D. E. Kieras and S. Bovair, "The role of a mental model in learning to operate a device." Tuscon, Arizona: University of Arizona, Report No. 13, 1983.

[11] J. Rasmussen, "On the structure of knowledge--a morphology of mental models in a man-machine system context." Roskilde. Denmark: Riso National Laboratory, Riso-M-2192, 1979.

[12] H. Stassen and R. J. van der Veldt, "Human operator models: a useful tool in MMIF design?" Delft, The Netherlands: Delft University of Technology, Laboratory for Measurement and Control, 1982.

[13] R. D. Smallwood, "Internal models and the human instrument monitor," IEEE Transactions on Human Factors in Electronics, vol. HFE-8, pp. 181-187, 1967.

[14] J. Rasmussen, "Outlines of a hybrid model of the process plant operator." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[15] R. E. Curry and E. G. Gai, "Detection of random process failures by human monitors." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[16] J. Rasmussen, "Models of mental strategies in proces plant diagnosis." In J. Rasmussen and W. B. Rouse (Eds.), Human Detection and Diagnosis of System Failures. New York: Plenum, 1981.

[17] T. B. Sheridan, "Toward a general model of supervisory control." In T. B. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[18] J. Rasmussen, "Skills, rules and knowledge; signals, signs. and symbols, and other distinctions in human performance models," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, pp. 257-266, 1983.

[19] R. E. Hall, J. R. Fragola and W. J. Luckas, Jr. (Eds.), The 1981 IEEE Standards Workshop on Human Factors and Nuclear Safety. New York: Institute of Electrical and Electronics Engineers, 1981.

[20] L. P. Goodstein, "Discriminative display support for process operators." In J. Rasmussen and W. B. Rouse (Eds.), Human Detection and Diagnosis of System Failures. New York: Plenum. 1981.

[21] R. J. Braune and S. R. Trollip, "Towards an internal model in pilot training," Aviation, Space, and Environmental Medicine, vol. 53, pp. 996-999, 1982.

[22] A. Shepherd, E. C. Marshall, A. Turner and K. D. Duncan, "Diagnosis of plant failures from a control panel: a comparison of three training methods." Ergonomics, vol. 20, pp. 347-361, 1977.

[23] E. R. F. W. Crossman and J. E. Cooke. "Manual control of slow-response systems." Paper presented at the International Congress on Human Factors in Electronics, Long Beach, California, 1962. Also in E. Edwards and F. P. Lees (Eds.), The Human Operator in Process Control. London: Taylor & Francis, 1974.

[24] B. W. Surgenor and J. D. McGeachy. "Validation for performance measurement in the task of fault management," Proceedings of the 27th Annual Meeting of the Human Factors Society, pp. 1058-1062, 1983.

[25] N. M. Morris, W. B. Rouse, and J. L. Fath, "Studies of human problem solving in process control: Evolution of an experimental simulation." Submitted for possible publication in IEEE Transactions on Systems, Man, and Cybernetics.

[26] J. L. Fath, "Design and evaluation of a process plant simulator." Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, Report No. 82-3. 1982.

[27] W. L. Williams, Jr. and P. G. Whitmore, Jr., "The development and use of a performance test as a basis for comparing technicians with and without field experience: the Nike Ajax maintenance technician." Washington, D. C.: George Washington University. Human Resources Research Office. Report No. 52, 1959.

[28] W. B. Rouse and S. H. Rouse, "A framework for research on adaptive decision aids." Norcross, GA: Search Technology, Inc., Report No. 83-1, 1983.

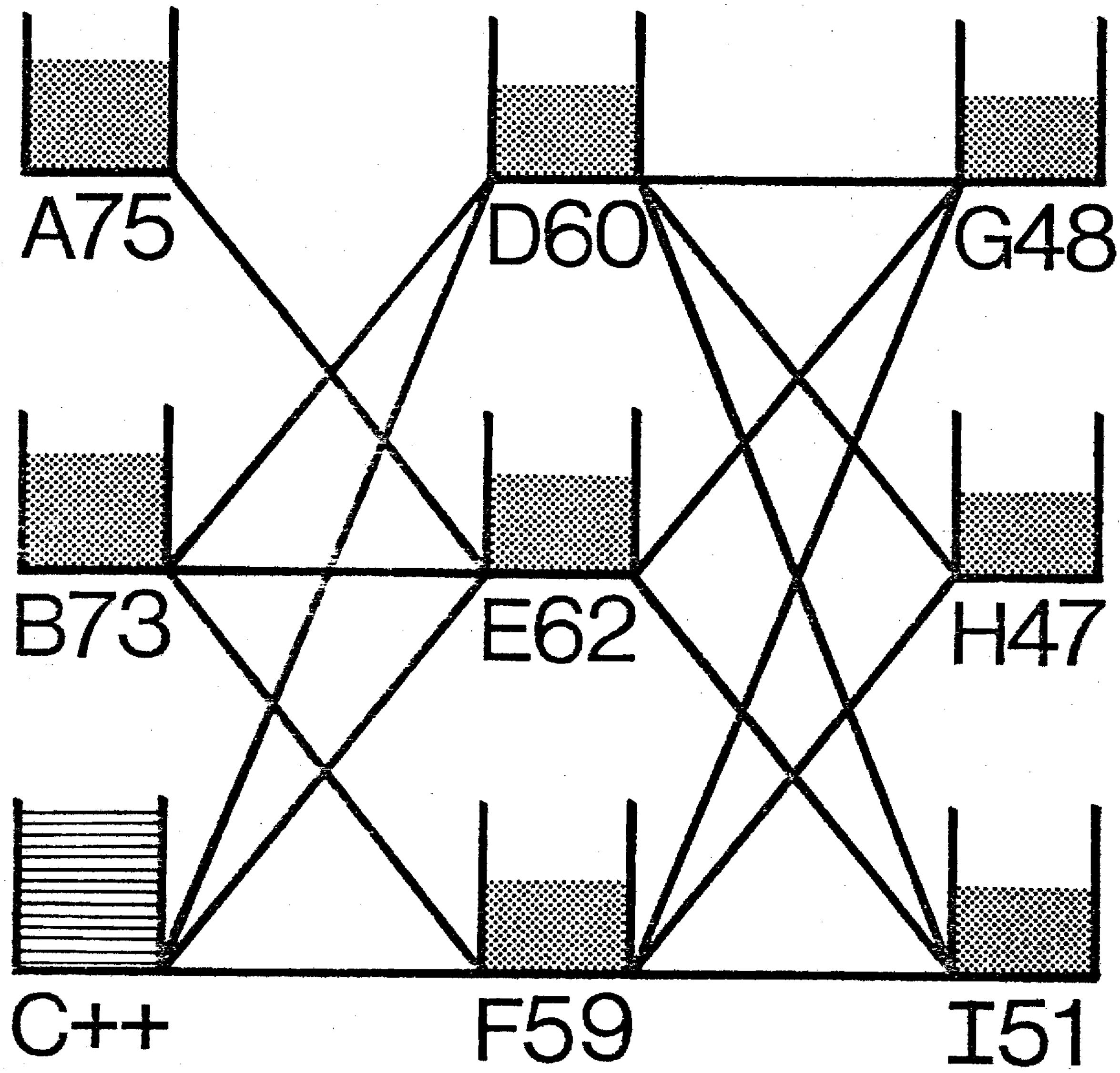


Figure 1. Sample graphic PLANT display.

```
Time = 183                                    Total Production =    88476.0
Avg. Height = 64.33                           Current Input =  540.0  pi = 180.0
Resources = 52,110                            Current Output = 540.0  po = 180.0
Limit Alarms:  a b c d e f g h i
Your Action = _

-------------------------------------------------------------------------------
 Time      Action      Message
-------------------------------------------------------------------------------

 182       rpc         Repair crew dispatched to  pump c

 181       afc         Result of flow tests:  0.000  0.000  0.000

 177       sk5

 176       otf

 175       cve.h

 174       po180
```

Figure 2. PLANT information display.

Table 1

Correlations Between Dependent Measures

	PROD[a]	TRIPS[b]	NOPEN[c]	VAR[d]	FIX[e]	TEST2[f]	SECT1[g]	SECT2
TRIPS	-.437*							
NOPEN	.673*	-.706*						
VAR	-.574*	.967*	-.768*					
FIX	-.429*	.141	-.234	.218				
TEST2	.191	-.200	.313	-.258	.107			
SECT1	-.021	.261	-.189	.268	-.100	.148		
SECT2	.190	-.238	.366*	-.292	.225	.860*	.040	
SECT3	.105	-.161	.157	-.197	-.056	.661*	-.022	.225

[a]PROD = average production/iteration.

[b]TRIPS = number of automatic valve trips/iteration.

[c]NOPEN = average number of valves open/iteration.

[d]VAR = variance of tank heights in PLANT.

[e]FIX = average time to diagnose valve and pump failures.

[f]TEST2 = overall score on Test 2.

[g]SECT1, SECT2, SECT3 = scores (% correct) on subsections of Test 2; SECT1 = minimal questions, SECT2 = procedural questions, SECT3 = principles questions.

*$p < .05$

CENTER FOR MAN-MACHINE SYSTEMS RESEARCH
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
ATLANTA, GEORGIA 30332 (404) 894-3996

March 30, 1984

NASA Scientific and Technical
Information Facility
Post Office Box 8757
Baltimore/Washington International Airport
Baltimore, Maryland 21240

Gentlemen:

Enclosed please find two copies of the Semiannual Report for NASA Grant No. NAG-2-123 for the period of September 1983 - February 1984. The copies include the original plus one photocopy.

Sincerely,

William B. Rouse
Professor and Director

cc: E. Palmer
University Affairs Office
F. Cochran

Enclosure

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

September 1983 - February 1984

William B. Rouse, Principal Investigator

John M. Hammer

Nancy M. Morris

Annette E. Knaeuper

Edward N. Brown

Charles M. Lewis

Teresa L. Mann

Wan C. Yoon

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology has resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Further, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering methods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments, particularly in terms of alternative approaches to training and aiding. Both of these efforts are producing results that are planned to be tested further in the Center's evolving full-scope flight simulation facility. Progress on developing this facility is discussed in the section on the intelligent cockpit.

THE INTELLIGENT COCKPIT

Previous work includes a computer program which locates human errors in following procedural checklists. This program contains an artificial intelligence based model (scripts) of checklist execution. This internal model is kept current with the aircraft state and the flight crew activities. Errors are detected from discrepancies between the model and the flight crew's actions. The program was evaluated on data from an earlier experiment on a link GAT-II simulator. The program was able to identify practically all known errors plus several additional ones.

SECOND GENERATION INTELLIGENT COCKPIT

The capabilities of the existing script-based aid will be expanded into two areas: script power and qualitative models. Script power itself can take two forms. First, the intelligent cockpit should always be following the same checklist as the flight crew. In other words, it should look at the same cues that an expert observer would. This has not been extensively tested in the existing program.

A qualitative model based on physics primitives (force, friction, rotating masses, heat) would offer two new capabilities to the intelligent cockpit. The first is to offer explaining power to script checklists. The current scripts are limited to knowing what is to be done. There is no knowledge about what happens when, say, a particular checklist step is omitted. A qualitative model would be an appropriate way to represent improved script semantics.

Ultimately, these aiding concepts are to be tested in the Center's full-scope flight simulator. One experiment that is envisioned is to test the procedural aiding in an online environment. The current program, it should be recalled, analyzes data from completed simulator runs. The principal thrust of this experiment would be to evaluate different aiding strategies. In other words, once a procedural error is found, what strategy should be used to alert the flight crew.

SIMULATION SOFTWARE

The Center's full-scope flight simulator is being revamped for realistic experiments with computer-based decision aids. Three major tasks must be accomplished before experimental work can be done. First, simulation software must be written to mimic flight dynamics and navigation. Second, the existing controls must be located and interfaced to the simulation software. Third, the outputs from the simulation must be displayed on the new CRT-based flight deck. Wan Yoon has primary responsibility for these software developments.

An initial version of the flight simulation software is now running on the VAX 11/780. This software is being extended to include the following features. An autopilot that operates in both manual and command mode has recently been completed. The current task is to enhance the engine model from a simple throttle to thrust relationship to one that simulates thrust, engine pressure ratio, exhaust gas temperature, N1, N2, and full flow. The effects of air density, temperature, and airspeed will also be modeled.

A second set of enhancements will be necessary to use the program in experiments. Flight data must be recorded in a disk file and be replayed for offline analysis and debugging. The program must be modified to take inputs from the A/D converter and produce outputs for the displays. (The current version has simple terminal input and output.)

SIMULATOR HARDWARE

The most pressing task in the entire simulator project is to get the simulator hardware ready. Edward Brown has primary responsibility for this task. The current areas of concentration are the pedestal, yoke, and flight instruments. The basic tasks for the first two areas are to dissemble, locate sensors, and rewire/reassemble. Fortunately, more sensors have been found than was originally expected. The missing ones, typically an assembly sensing a number of controls, must be located from other sources or constructed in house.

The flight instruments are going to be provided on CRT displays. The hardware effort in this area is to mount the displays in the existing framework that once held electromechanical gauges. Of the equipment needed, one CRT and two keyboards have arrived. Four CRTs and the A/D conversion equipment have been ordered.

The status of these initial three areas is as follows. The pedestal and yoke has been disassembled. The sensor arrangements are understood, and when the missing sensor boxes are located, reassembly can begin. The flight instrument panel work can begin when the above is finished and the displays arrive.

The time schedule for demonstrating the simulator is as follows. An initial demonstration will be made during the summer of 1984. Flight instruments, engine displays, and limited navigation abilities will operate. A second demonstration in the fall of 1984 will feature more complete navigation, data entry and simulation utilization of navigation and center of gravity, utilization of numerous discrete switches, and simulation of various failures in engine, hydraulic, power generation, and similar systems.

STUDIES OF HUMAN PROBLEM SOLVING

In order to support domain-oriented projects such as the intelligent cockpit, it is necessary to increase our basic understanding of human decision making and problem solving. This has been a main tenet of this program of research since its inception and continues to be a guiding principle.

The latest efforts in the area of basic studies of human problem solving have focused on two topics. The first is the use of a model of human problem solving (KARL) as a means for online training and aiding in PLANT. The second topic is the development of identification methods for rule-based models.

A MODEL-BASED APPROACH TO ONLINE TRAINING AND AIDING

The results of Nancy Morris' Ph.D. thesis, which were summarized in the last report, indicated that subjects who operated PLANT were not always aware of <u>when</u> to apply the knowledge that they gained during training. The results of Annette Knaeuper's M.S. thesis, supported under an Office of Naval Research grant, illustrated systematic inconsistencies between what subjects knew and said they would do, and what they actually did. This assessment was made by developing a rule-based model of problem solving (KARL) that essentially received the same instructions as subjects. Comparing the model's choices of actions to those of subjects uncovered the inconsistencies.

These conclusions were further supported by the recent M.S. thesis of Teresa Mann. She performed a fine-grained analysis of subjects' compliance with PLANT's procedures. An interesting conclusion was that the group with the most training, including basic principles of PLANT operation, appeared to follow the procedures least. In general, subjects did not follow procedures to the extent that their training would have led one to expect. It may have been that subjects did not always know when procedures were applicable.

Online Training and Aiding

The fact that KARL could be used to analyze the sequence of subjects' actions led to the idea of using KARL as an online "coach" for both training and aiding. During this reporting period, KARL has been modified (under ONR support) to operate in this manner and an experiment with PLANT has been planned (under NASA support) to evaluate the concept.

KARL provides three types of online assistance. First, KARL provides a context-specific situation assessment as well as a recommended procedure (i.e., none required, normal tuning, particular procedure applicable, and none applicable). KARL also provides performance monitoring by assessing the consistency of subjects' actions relative to applicable procedures and providing context-specific feedback and explanations. Finally, KARL provides performance feedback by informing subjects if instability problems are improving, excessive, or extreme.

Experimental Plan

To evaluate the use of KARL in this manner, a group of eight subjects will be run as a fifth group in the original experimental design of Nancy Morris. This group will receive the Procedures and Principles training developed by Morris (group D in the original design) as well as online assistance via KARL.

Situation assessment and performance feedback will be provided during sessions five through twelve. However, performance monitoring will only be provided during sessions five through eight. This will allow a determination of whether or not consistency is maintained by subjects once monitoring by KARL ceases. A 13th and final experimental session will be performed without any assistance from KARL to assess whether or not performance degrades without any aiding.

The plan is for this experiment to be performed in April.

IDENTIFICATION METHODS FOR RULE-BASED MODELS

Rule-based models have served well as explanations of problem solving and, as implemented in KARL, have shown performance similar to human subjects. One difficulty with this approach, however, has been the tedious and subjective process of manually identifying rules. A methodology is now being refined for algorithmically identifying relations in data in a form compatable with these models. Mike Lewis is performing this research in conjunction with his Ph.D. thesis in psychology.

Any inference from observations requires some model, explicit or implicit. Linear models such as analysis of variance or regression have proved the most popular for data analysis. In these models the dependent variable is treated as an amalgam of effects contributed by the independent variables. When the dependent variable is continuous powerful tools are available which disentangle these influences. While sufficing to describe many observed relations, others appear to be of a qualitatively different form. In events such as crystal formation or catalytic reactions the characteristic of greatest interest is a quantum change in state attributable to a synthesis of variables. For such relations a more suitable description would be specification of conditions under which this change occurs. This specification may be considered a rule, as it assigns an action (change in state) to a range of conditions.

In problem solving a subject commonly has a variety of available actions. His task lies in choosing a sequence consonant with some goal. The saliency of variables will often depend on their context. For example, functioning headlights may be a useful cue in diagnosing an electrical fault yet be irrelevant to fixing a flat tire. This dependence on context and qualitative differences among responses are hallmarks of synthetic relations in which the event depends on the coincidence of values rather then individually contributed effects.

If a response may be to be controlled by context, an appropriate model must identify the relation between contexts and responses rather than the influence of particular variables. The context of an observation is defined by the values of its variables which are coordinates locating it in an event space. Context dependent relations can be expressed directly in terms of these n-tuples. The multidimensional set of independent variables used by linear models can be replaced by a single unordered context dimension with constituent variables serving to identify its values. Description of the observations in more parsimonious and general form requires identification of regions of this event space for which this effect of context is equivalent. Partitioning the event space into such regions results in a set of situation-action rules describing consistencies in behavior.

For example: If pilots were observed to maneuver only when an intruder was 'too-near' horizontally and vertically three rules of this type would result.

1. horizontal distance is too-near &
   vertical distance is too near -> maneuver

2. horizontal distance is near or far &
   vertical distance is too-near,
   or near, or far -> No maneuver

3. horizontal distance is too-near,near,or far
   & vertical distance is near or far -> No maneuver

Identification of such rules presents a combinatorial problem of finding regions associated with particular responses. In this example there are 18 possible observations (i.e., 3 horizontal distances x 3 vertical distances x 2 responses), which may be described by 68 possible rules. As the size of the problem increases these numbers rapidly become intractable. In the case of four explanatory variables and a dependent variable each taking on ten values there are one million possible observations and between 10**8 (ordinal domains) and 10**17 (nominal domains) possible rules. Applicable statistical methods would require ennumeration of hypotheses (rules) and evidential weighting for observations.

Fortunately alternative nonstochastic methods have been developed. Over the past fifteen years researchers in "machine learning" have experimented with methods based on the predicate calculus which draw consistent generalizations from sets of examples. The most familiar of these is Winston's (1970) "Blocks World" program. These generalization procedures are of two basic types: those which find the most general expression which discriminates between positive and negative instances and those which find the most specfic expression covering a set of positive examples. Some programs such as INDUCE 3 (Michalski,1980), and MetaDendral (Mitchell,1977), employ both approaches.

A further distinction lies in the acceptable representation of observations. Conventional methods of analysis and some pattern generalization approaches are limited to attribute spaces. In an attribute space context is defined by global properties of the observations which can be represented as measurements. Pattern generalization is not in principle restricted to such spaces. If we were to describe rooms in an attribute space we might arrive at the expression: room={ (walls>2),(ceiling=1),(floor=1)}. In this case we could not tell a room from a stage or some random assembledge of room parts. A structural representation allows description of relations : room={ Joined(wall(x),wall(y)) and On-top(wall(x),floor) and On-top(ceiling,wall(x))}. In this example the description, On-top(wall(x),floor) represents a generalization of the observation, On-top(wall(1),floor) and On-top(wall(2),floor) and... In principle elementary structural relations such as On-top(wall(1),floor) can always be represented as attributes, however, generalization among them cannot be performed. In complex domains such as problem solving this sort of data reduction may be crucial.

For example, In the PLANT simulation there are multiple columns of tanks. A good rule for locating a failed valve is to find an increasing level in the tank on its left accompanied by a decreasing level in the tank on its right. Finding this rule requires simultaneous generalization of structural relations (column-left, column-right) and attributes (increasing-level, decreasing-level).

Interpretation of Rules

While the rules produced by generalization programs bear a formal resemblence to production systems models of cognitive processes the two should be distinguished. Theory based models such as KARL make strong assumptions about the ways in which decisions are made (control structures). These models attempt to describe not only what a person does but how he does it. If accurate, fitting such models will provide insight into human capacities and limitations and be generalizable to other tasks.

Despite the rubric of "machine intelligence" pattern generalization algorithms are as purely computational as least squares. The rules abstracted embody only consistencies among contexts and responses. A production system composed of such rules emulates rather than models the cognitive processes of a person performing the task.

There are, however, connections between the two types of rule. Markov(1955) has demonstrated that any realizable algorithm can be re-expressed as an equivalent (in behavior) set of production rules without control structure. This property may be used to reduce algorithms of differing structure to comparable form. Pattern generalization can be used to identify behavioral 'algorithms' at this lowest common denominator providing a metric for comparing cognitive models and observations.

In data analysis the property of emmulation, alone, suffices. By describing associations between contexts and decisions rather than performance summaries, contexts producing anomalous decisions are readily identified. As with identification of interactions in linear models the resolution with which rules may be specified is dependent on the extent to which combinatorial possibilities are represented in the observations. Differences in behavior attributable to a manipulated variable are indicated by its appearance on the condition side of a rule. These differences may be of two types. If a rule employs different variables it may be interpreted in terms of a shift in the saliency of cues (change in rules). Differences in ranges of values may be attributed to changes in perceptual precision.

Evaluating Rules

Since identification of "rules" is based on logical consistency rather than stochastic considerations the inconsistencies of actual observations may cause difficulties. The discriminant procedures are typically limited to rules which exclude negative examples while the nondiscriminant methods lack a means of detecting abberations. This problem of bias reduction can be ameliorated by aggregating observations. Inconsistencies in responding to a context may be resolved by entering the modal response or a new category formed by the disjunction or the most prevalent responses. In the earlier example disjunction of the response might have resulted in a rule such as:

horizontal distance is near & Vertical distance is too-near -> maneuver or No maneuver.

In this case consistent inaction has been observed only when the intruder was not "too-near" in either dimension and consistent maneuvering only when both were "too-near". The inclusion of a disjunctive response allows specification of conditions under which no single response is consistenly produced. This becomes important when there are a number of consistently chosen alternatives which form a proper subset of responses available.

While identifying consistent relations in observations pattern generalization lacks the inferential machinery available to parametric identification. Once identified, however, the system of rules may be evaluated in a parametric model. If the system of rules and responses are self contained, inconsistencies observed can be attributed to disturbances from outside of this system and are therefore independent of the contexts identified by the model. If response categories are treated as a logit vector the rules specify the cells within a contingency table associated with consistent responding. A chi-square test of independence for the entire table consisting of contexts and responses provides a test for a null hoypthesis of no effects. A subsequent test of quasi-independence for the incomplete table excluding the cells specified by the rules provides a goodness-of-fit test of the rule model.

Additional statistics are available to evaluate the performance of rules relative to the variation among observations. Goodman and Kruskal's Proportional Reduction in Error (lambda) statistic measures and supplies confidence intervals for the relative improvement in prediction attributable to a rule. A second measure, t, (Margolin and Light,1974) may be used to find the proportion of variance explained by a rule or rule set and test its significance.

Induce 3

A pattern generalization program, INDUCE 3 (Hoff,et al.,1983),has been obtained for use in rule identification. INDUCE 3 finds discriminant or characteristic generalizations of examples. It is suitable for both attribute only and attribute plus structure event spaces.

As with other predicate calculus based generalization programs, INDUCE performs structural generalizations using graph search. By representing observations as a directed graph with labeled edges the generalization problem becomes one of finding a maximal common subgraph among examples. Two mechanisms for generalization are immediately available as a result of this device. If at least one of a node's edges is not common among examples, the node is absent from the common subgraph and its condition is dropped from the generalization. Matching subgraphs involving same type nodes (walls in the room example) provides a "turning constants into variables" facility, forall x(On-top(ceiling,wall(x))). This generalization method is less well suited for dealing with attributes which can be affected only by dropping their condition.

INDUCE overcomes this deficiency by incorporating a second algorithm for attribute generalization. In INDUCE structural generalizations ignoring attributes are initially obtained through graph matching. The resulting generalizations are appended to the examples as attributes. The Aq algorithm, operating in an event space similar to that described earlier, produces consistent structural plus attribute generalizations by finding the intersection of positive examples with the complement of the negative examples.

Data analysis using pattern generalization is made possible by the efficiency of this algorithm. The methods based solely on graph matching must process examples serially, frequently backtracking to mend inconsistencies. The Aq algorithm achieves economy by considering the constraints imposed by the examples simultaneously. An added advantage of this algorithm is its ability to generalize through internal disjunction (i.e., horizontal=too-near or near), and to uncover consistencies involving combinations of structures and attributes.

CDTI

As an initial step, these techniques are being applied to a single decision task which can be adequately represented by attributes alone. Data from an experiment conducted by Palmer (1983) investigating the effects of information quality and intruder characteristics in the use of a cockpit traffic display is being re-analyzed.

Sixteen pilots "flew" simulated encounters under three display conditions. Pilots were instructed to maintain a steady course, using the autopilot unless they received a threat advisory. In response to the threat they were to maneuver to maintain a horizontal separation of greater than 1.5 nm and a vertical separation in excess of 500 ft. They were advised that an appropriate strategy was to maneuver so that the intruder would pass further away but in the same orientation at the point of closest approach.

In the least informative condition the display protrayed the relative positions of the ownship and the intruder along with tags showing their altitudes. The predictive display provided ground referenced predictors showing predicted positions of the ownship and intruder as well as a tag showing the intruders projectd altitude at time of closest approach. In the third condition noise was introduced into the predictive display

Examples from an initial analysis using only programmed horizontal miss and intruder vertical velocity measurements to define the contexts may be used to illustrate the technique. In these generalizations pilot responses were examined in terms of both single response dimensions and combinations of dimensions.

In both treatments of the response a strong link was found between a constant intruder altitude and a vertically away maneuver. When the response combination, vertical-away, no horizontal action, is generalized the rule below results:

Intruder vertical velocity is constant
-> vertical-away/horizontal-inaction

This rule correctly matches 155 of the 768 events while resulting in 38 errors. This corresponds to 13% of the variation in the responses and is significant at the .01 level.

When response dimensions were considered independently a somewhat different version indicating an influence of the displays was found:

Display is predictor or predictor+noise & programmed
horizontal separation is not 1 nm & intruder vertical velocity
is zero or 1000 ft/sec -> vertical-away

This rule correctly matches 115 of the 768 events producing 15 errors. In the encounters presented in the study, non-zero vertical velocities usually resulted in reversal of the aircrafts' relative vertical orientation at the point of closest approach. This would make a vertically away response appropriate primarily for intruders approaching at a constant vertical velocity. The second rule expressing this strategy is restricted to pilots using predictive displays. Pilots flying without the data tag showing intruder altitude at point of closest approach were observed to take the same action under a variety of inappropriate circumstances.

These rules illustrate an instance in which the difference between conditions cannot be attributed to strategy since in the constant altitude encounter all pilots consistently maneuvered in accordance with their instructions. Despite their utilization of the proper strategy, pilots flying without the data tag showing intruder altitude at point of closest approach failed to estimate that quantity reliably when the intruder's altitude was changing.

A second example shows advantages for predictive displays in avoiding unneccesary maneuvers.

Display is predictor or predictor+noise &
programmed horizontal miss is 2 nm
-> vertical inaction/horizontal-inaction

This rule correctly matches 33 events while erring in 3 instances. Use of this rule results in a 6% reduction of error in describing the observations and accounts for 3% of their variation. Under these conditions the intruder will pass well beyond the 1.5 nm at which a maneuver might be required. In this case interpretation becomes equivocal. It is not clear whether pilots lacking predictive displays fail to recognize the benign nature of these encounters or have adopted a generalized strategy of maneuvering in all encounters.

Another rule indicates other ambiguities which may occur if rules are interpreted solely on the basis of their "performance".

Programmed horizontal separation is less than 2 nm &
passing position will intercept or is in front &

intruder vertical velocity is non zero
-> horizontal-toward/vertical-inaction

This rule correctly matches 122 events but is in error for 118 others. None the less it results in a 15% reduction in error and accounts for a significant 5% of the variation in the responses.

An explanation lies in the nature of the encounters described. Under the avoidance instructions neither horizontal direction is preferred when trajectories are intersecting. Similarly the choice of a vertical or combination of horizontal and vertical maneuvers might be equally acceptable. For the intruder passing in front at an angle which is not obtuse, however, the vertical-toward maneuver is preferred as it will cause the intruder to pass further in front. This rule indicates that in a substantial (16%) number of encounters the horizontal-toward/vertical-inaction response was chosen when it was a preferred response or one of a number of appropriate responses. If disjunctive response alternatives had been available this paradox of simultaneous high predictability and error rate might be resolved.

Future Plans

More exhaustive analysis of the CDTI data employing additional variables is planned. The data from the PLANT simulation described earlier in this report will be used to examine the effectiveness of these rule identification techniques in dynamic decision making in an attribute plus structure space. In addition to greater complexity the nature of the task presents difficulties of theoretical interest. Changes in goals represented by situation-situation rule control structures are widely employed in problem solving research. For a pattern generalization algorithm identification of these rules is an unobserved state problem. The use of states supplied by KARL or collection of verbal protocols are being considered to confront this problem. The appropriateness of a single response under disparate conditions represents an additional difficulty when operating in an attribute plus structure space due to the heuristic nature of INDUCE's graph matching algorithm. Solutions to these problems will be examined in this research.

It is hoped this research will result in the availablility of rule-based methods for analyzing behavioral data and help establish closer ties between normative rule-based models and experimental data.

References

1. Hoff, William A., Michalski, Ryszard S., Stepp, Robert E. "INDUCE 3: a program for learning structural descriptions from examples", in draft form, Department of Computer Science, University of Illinois at Urbana-Chanpaign, 1983.

2. Margolin, B.H. and Light, R.J. "An analysis of variance for categorical data II: small sample comparisons with chi-square and other competitors". Journal of the American Statistical Association, 69,1974.

3. Markov, A. A. Theory of Algorithms. Moscow: Academy of Sciences of the U.S.S.R., 1955.

4. Michalski, Ryszard S. "Pattern recognition as rule guided inference", IEEE Transactions on Pattern Analysis and Machine Intelligence, 4,1980,349-361.

5. Mitchell, T. M. "Version spaces: a candidate elimination approach to rule learning", Proceedings of the Fifth International Joint Conference on Artificial Intelligence, IJCAI, Cambridge, Mass.,1977,305-310.

6. Palmer, Everett "Conflict resolution maneuvers during near miss encounters with cockpit traffic displays",Proceedings of the Human Factors Society-27th Annual Meeting,Human Factors Society, Santa Monica, Ca.,1983,757-761.

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Report

March 1984 - August 1984

William B. Rouse, Principal Investigator

John M. Hammer

Nancy M. Morris

Annette E. Knaeuper

Edward N. Brown

Charles M. Lewis

Wan C. Yoon

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

Increased requirements for safety and efficiency as well as increased availability of reliable and inexpensive computer technology have resulted in a trend of more and more computers being employed in flight management. However, this trend by no means indicates that human operators will disappear from aircraft cockpits. Instead, it means that the roles of the pilot, copilot, and flight engineer will evolve to include increased responsibilities for monitoring and supervising the various computer-based systems employed in the aircraft.

While this assessment of the future roles of the members of the flight crew is fairly easy to accept, it is certainly not straightforward to decide how various flight tasks should be allocated among humans and computers. Further, it is not clear how humans and computers should communicate regarding the process by which their tasks are performed and the products that result. This report discusses progress of a research program whose overall objectives include providing at least partial answers to some of the questions surrounding these issues.

The following two sections discuss two project areas which are currently being pursued in this program of research: 1) the intelligent cockpit and 2) studies of human problem solving. The first area involves an investigation of the use of advanced software engineering methods (e.g., from artificial intelligence) to aid aircraft crews in procedure selection and execution. The second area is focused on human problem solving in dynamic environments, particularly in terms of identification of rule-based models and alternative approaches to training and aiding. Both of these efforts are producing results that are planned to be tested further in the Center's evolving full-scope flight simulation facility. Progress on developing this facility is discussed in the section on the intelligent cockpit.

THE INTELLIGENT COCKPIT

Design goals for the intelligent aid are in the following section. A review of progress in developing the DC8 flight simulator is in the second section.

GOALS FOR ANALYSIS OF DATA IN THE INTELLIGENT COCKPIT PROGRAM

The Problem

The data are a large, rich state vector of the aircraft. The intelligent aid is to monitor this data to watch for pilot checklist error and unsafe conditions either present or future.

The current approach to monitoring has been to divide the flight into phases according to rules that examine the data past and present. It is possible to be more specific about a phase than about the whole flight. Thus, it is possible to apply pre-programmed checks as appropriate to that phase. While this approach has merits, there are design limits to what can be preprogrammed. In other words, this approach works only for the situations the designer anticipates. In particular, this approach can be made to look quite good in a controlled experiment where the intelligent aid has been programmed to aid on those situations the pilot will encounter. Our approach to demonstrating intelligent aid concepts is to build something that at face value will handle a wide variety of situations. If, due to complexity, a complete aid cannot be constructed, we would prefer an in-depth aid for a particular problem (e.g., electrical malfunctions). We reject an aid that is able to catch a few problems of all kinds but which appears inadequate for complete coverage.

External Goals

The external goals are the visible product that an observer or pilot would see when the program is running. The following are planned.

1. The program will drive two displays. The first will be the experimenter - system designer display. It will be used for program debugging, etc. The second will be a display for the pilot. It will be created even if there is no pilot to observe it (e.g., off line simulator data). This generality will allow the aid to be incorporated into the DC8 flight simulator without substantial reworking. It will also allow subjective pilot evaluation before being placed in the simulator.
2. The pilot's display will feature the following. Procedures will automatically be selected for display by the aid. When the aid detects an error, the procedure display will be changed, perhaps by highlighting.
3. The display will alert the pilot to aircraft operation outside the normal regime at the present time and in the future.

Internal Goals

The internal goals describe how the program is organized. Some of these supplement the external goals; some are not apparent from that vantage point.

1. The program will understand the procedure in terms of a model of the aircraft. This model will be used to predict the future states of the aircraft. The program will understand the effect of procedural errors, not just know how to detect them as was the case in the earlier version of the program.

2. To the maximum extent possible, the program will generally apply to all commercial transport aircraft. This will facilitate changing from NASA's B727 to our DC8.

3. Existing scripts will be enhanced and will represent more than a procedure checklist. The script will be active whenever the aircraft is in the appropriate state. While the aircraft is in the script, constraints on safe flight will be constantly monitored.

4. The program's goal will be to avoid aborting the flight plan. This goal is made up of a number of subgoals which are by aircraft subsystem. For example, to avoid aborting the flight plan, the following failures must be avoided

   propulsion failure
   aerodynamic failures
   hydraulic failures
   etc.

   Each of these areas can in turn be broken down into finer problems

   propulsion failure
   - over stress engines
   - engine ice up
   - engine stall
   - engine oil pressure
   - engine explosion and airframe/airfoil damage
   - etc.

   To be systematic, it would be important to enumerate as many forms of failures as possible. Organizing them by subsystems would tend to keep the designer from missing a failure mode. To restrict the aid program to a manageable size, it may be necessary to aid only a subset of the potential failures (e.g., only electrical problems). This would, however, demonstrate the approach's generality, and it would be clear how to generalize to the entire aircraft.

5. The program will be coded as a rule-based system in LISP. As we have stated before, the problem of intelligent aiding is principally a symbolic problem. In addition, the debugging tools for LISP are far superior to those of other languages. Also, there is a practical advantage to the use of LISP. If the simulator code or the intelligent aid should end up consuming too much of the VAX's capability, it should be possible to move the LISP aid code to a LISP machine. This should be considerably cheaper than other VAX.

6. The program must make some predictions about the aircraft's future. This is necessary to check for safe, future states. Representing and manipulating the passage of time is a recognized problem in artificial intelligence. Hopefully, some contribution can be made to this area.

Conclusion

The proposed new aid will have an improved understanding of the aircraft and consequently the ability to alert the pilot to present and future dangers. The aid will also be useable on both the B727 and the DC8.

SIMULATOR DEVELOPMENT

Simulator Hardware

Most of the switches and controls (overhead panels and flight controls) have wires connected that go outside the cockpit and are waiting to be connected to the A/D converter. A few engine controls and the elevator controls must have sensors designed and installed before being wired.

All of the CRT displays have been mounted in the flight deck. A cowling must be added to enclose one of the CRT's forward of the pedestal. Both this cowling and a force feel system for the control yoke are being designed and built in the GTRI shop. The custom keyboards have been designed and are currently being fabricated. A force feel system for the pedals is installed and undergoing final adjustment.

Simulation Software

The simulation software currently supports high fidelity engines, flight dynamics, radio navigation and an autopilot. Hydraulic, electrical, and fuel systems are not being simulated in any but the most elementary ways in this version of the simulator. The remaining work on the simulation software is to integrate it with the display software and to modify the takeoff routines to use flight dynamics. Some minor modifications (coefficient changes and term changes) are being made to the flight dynamics to make it perform as a commercial transport.

Display Software

There are three displays to be produced for in the simulator: the flight instruments, the engine status, and the navigation/autopilot/communication display. The engine status display is being programmed now; it is fairly simple. The navigation/autopilot/communication display specifications have been worked out and will be programmed after the engine display is completed.

The flight instrument display is by far the most complicated display. The original plans were to use an Apple II to drive the color displays. It appears this will not work because the Apple is not fast enough. The following simple analysis shows how this was determined. The ADI must be updated frequently; at a minimum, updating this display requires drawing of 11 or 12 lines. Timing estimates show the Apple (using Pascal) can erase an old line and draw a new line in .02 + .0002*pixels seconds. Thus, the ADI graphics can be redrawn about once every .250 seconds. In practice, the Apple II will be much slower than this, since it would have to make computations, receive VAX input, and display all of the flight instruments, not just the ADI. Thus, a 4 Hz ADI bandwidth is quite optimistic with regard to Apple II capabilities, but it is insufficient with respect to the pilot's needs. It is our understanding that a 6 to 8 Hz bandwidth is necessary for realistic control.

While it may well be possible to write an assembly language program to make the Apple perform as desired, it would be expensive from a labor standpoint. Consequently, alternative graphics devices are being investigated. Currently, the specifications for this graphics device are that it have a high speed, parallel interface to the VAX and a display list. The parallel interface is necessary to update the display quickly. A display list is necessary so that all the dynamic elements on the screen can be changed in real-time.

It has also become apparent that seven more terminal ports will be necessary to drive the simulator displays. Three of these ports will be for output, three for input, and at least one will be needed by the experimenter to control the simulation. We would prefer to add more ports rather than take away from those we already have. Fortunately, ports cost only $200 each in groups of sixteen.

HUMAN PROBLEM SOLVING

As noted in the Introduction, research in this area is focusing on: 1) identification of rule-based models and, 2) alternative approaches to training and aiding. Both of these efforts are oriented toward human problem solving in dynamic environments including aviation and process control.

Mike Lewis, in conjunction with his Ph.D. thesis, is pursuing identification of rule-based models. The goal is to lessen the usual substantial subjectivity in the formulation of rule-based models by developing and testing algorithmic approaches. In this report, Mike discusses his use of such approaches for analysis of the CDTI data of Ev Palmer and his colleagues.

During the past three years, a portion of the efforts in problem solving have been focused on process control. Using a simulation called PLANT, Nancy Morris investigated the effects of types of knowledge on human performance. Annette Knaeuper developed a rule-based model of human problem solving in PLANT. In comparing the behavior of this model with that of humans, Annette found that humans often did not follow their instructions, namely, PLANT operating procedures. This led to the idea of using the rule-based model for online aiding and training of operators. Annette and Nancy performed an initial empirical evaluation of this concept, the results of which are discussed in this report.

IDENTIFICATION OF RULE-BASED MODELS

Charles M. Lewis

INTRODUCTION

Researchers investigating pilots using cockpit displays of traffic information (CDTI) (Palmer et al., 1980 and Smith et al., 1982) have found choice of control action to depend upon individual differences as well as encounter or display characteristics. In the development of the CDTI it is important that both generalized strategies shared by all pilots and idiosyncratic choices of the few be understood. As the CDTI will likely be used in conjunction with a collision avoidance system (CAS) it is important to gauge the influence of the display on pilot maneuvers so that advisories can be formed which are both consonant with pilot strategy and avoid conflicts among strategies. The radar assisted collisions discussed by Curry (1972) demonstrate the danger of introducing such technology without consideration of operator strategies.

While established policy capturing methods exist for examining the influence of variables on decisions, they fail to elucidate what the influenced policy actually was. The present work employs pattern generalization techniques to identify a production system of rules capturing the consistencies in observed behavior. A production rule consists of two parts, a set of conditions and an action. If the conditions are satisfied the rule's action is performed. If conditions are expressed using propositional logic, descriptions are restricted to attributes (global properties of an object). Measurements commonly used in science such as height, weight, or velocity are of this type. In such cases the conditional part of the rule defines a region of the attribute space within which the rule is true (responses are of the type specified in the action part of the rule). This representation proves convenient for visualizing set theoretic relations among rules.

While this formulation of pattern generalization is similar in approach to discriminant analysis there are some important distinctions:

1. A rule describes an enclosed region of the event space rather than a partition dividing the space into two parts.

2. More than one rule may be needed to describe a response if regions in which it occurs are separated.

3. Rules may vary both in generality (size of region) and selectivity (accuracy of discrimination).

This report describes an application of pattern generalization to identification of pilot strategies.

CDTI DATA

Data from an experiment by Palmer (1983) investigating the effects of information quality and intruder characteristics in the use of a cockpit display of traffic information instrument has been reanalyzed using pattern generalization techniques.

In this experiment Sixteen pilots "flew" sixteen programmed encounters under three display conditions. Pilots were instructed to maintain a steady course, using the autopilot unless they received a threat advisory. In response to the threat they were to maneuver to maintain a horizontal separation of greater than 1.5 nm and a vertical separation in excess of 500 ft. They were advised that an appropriate strategy was to maneuver so that the intruder would pass further away but in the same orientation at the point of closest approach.

In the least informative condition the display portrayed the relative positions of the ownship and the intruder along with tags showing their altitudes. The predictive display provided ground referenced predictors showing predicted positions of the ownship and intruder as well as a tag showing the intruders projected altitude at time of closest approach. In the third condition noise was introduced into the predictive display

EXPLANATORY VARIABLES

In this analysis encounter variables, describing the physical relationship between the intruder and ownship which the pilot is instructed to control, were differentiated from experimental variables. Five encounter variables were used. Four describe the relative positions of the aircraft at their point of closest approach as projected at time of alarm. The fifth measure, intruder vertical velocity, remains constant throughout the encounter.

hpass-horizontal passing position= behind, intercept, or infront.

hsep-projected horizontal separation= very near(0-.24nm), near(.24-1nm), or far(> 1nm).

vcross-vertically crossing trajectories= no,yes.

vsep-projected vertical separation= very near(0-140'), near(140-350'), or far(> 350').

vveloc-intruder vertical velocity= zero or non-zero.

Of the sixteen programmed encounters, encounters 7 and 8 which introduce crossing angle between the aircraft as a variable were excluded. Encounters 11-16 which involve abrupt changes in intruder course or introduction of intruder in near proximity to ownship, invalidating projections made at time of alarm, remain unanalyzed as in Palmer's (1983) report.

Two non-encounter variables were considered, display type and pilot. Display type in conjunction with the encounter variables describes the stimuli under which a decision is made. Inclusion of pilot identification in the generalization introduces individual differences.

RESPONSE VARIABLE

Pilots' responses were represented in terms of maneuvers toward or away from the intruder along a dominant axis. The dominant axis was determined by comparing the horizontal and vertical magnitudes of a maneuver to the respective tolerances which the pilots had been instructed to maintain. Five response classes result: no action, vertical-toward, vertical-away, horizontal-toward, and horizontal-away.

PERFORMANCE MEASURES

Nonparametric measures of association tau-b, the ratio of between groups sum of squares to total sum of squares, and PRE, the reduction in error relative to assigning the modal response to all cases, provide measures of rule performance which consider both coverage and discrimination. Tau-b provides a nonparametric analog to a squared correlation with values under .1 indicating a relatively weak association (corresponding to r < .30) and those over .5 a relatively strong one (corresponding to r > .70). Using tau, single rules are evaluated by comparing the distribution of response classes within the rule with that of the remainder of the cases. This provides a measure (barring intersections) of described variance contributed by that rule to its rule set. Rule set performance may be evaluated relative to the situations in which it applies or to the entire range of examples. When restricted to applicable regions, tau may be interpreted as a measure of the extent to which the rule set describes identified consistencies. When evaluated relative to the entire space, an additional "response category" formed by uncovered observations is required. In this case tau may be considered a measure of rule set performance relative to arbitrarily chosen examples.

Aq ALGORITHM

A pattern generalization program, INDUCE 3 (Hoff, et al. 1983), was obtained for use in rule identification. In this analysis only the VL1 (Aq) subprogram which identifies rules in propositional logic was employed.

The Aq algorithm generates a set of putative rules which match a particular positive example and exclude all negative examples. The rule which matches the most additional positive examples is retained. At each iteration sucessfully matched examples are removed from consideration. The process terminates when all non contradictory positive examples have been matched. Although previously matched examples cannot contribute to the retention of rules, they become "blanks" in the space, which being neutral, may become part of subsequent generalizations. The resulting rules may overlap substantially. If "rectangular rules" were identified for figure 1, three rules would be found: Rule-1=(2,3,4,5), Rule-2=(2,3,4,6,7,8), Rule-3=(1,3,7). As Rule-1 substantially describes this space with little non redundant contribution from rules 2 or 3, a parsimonious description may allow the smaller regions, 1,6,7, and 8 to go undescribed. Under other circumstances collapsing across an explanatory variable to produce a more general rule making occasional errors may be the choice dictated by parsimony.

EFFECTS OF NON-ENCOUNTER VARIABLES

The complete set of rules generalized using the Aq algorithm provides an upper bound on the consistency with which the responses can be associated with the explanatory variables employed. The contribution of a variable may be examined by comparing performances between rule sets generalized with and without that variable. While modest improvement will be obtained from an additional variable based on an increase in degrees of freedom, major improvements in description will mirror the "influence" of that variable on pilot decisions. An index to the relative contribution of an explanatory variable may be found by rank ordering rules by performance. The relative performance for same sized sets of rules can then be compared for rule sets of varying sizes.

A generalization based on encounter variables alone produced 230 correct matches with 154 errors resulting in tau-b=.18. If individual differences among pilots are considered as well, correct matches rise to 324 with 60 errors and tau-b=.61. Less improvement is found in the generalization based on encounter variables and display types: correct matches=254, errors=130, tau-b=.27. If display type and individual differences are entered into the generalization simultaneously only one error occurs yielding a tau-b of .99.

VARIABLES	No. Rules	Hits	FAs	Tau-b	PRE
ENCOUNTER	20	230	154	.18	.38
ENCOUNTER + DISPLAY	42	254	130	.27	.48
ENCOUNTER + PILOT	85	324	60	.61	.76
ENCOUNTER + PILOT + DISPLAY	114	383	1	.99	.99

Considering performance as a function of the number of rules reveals the same ordering of effects as found for the complete rule sets. The steeper slope of the generalization including individual differences and display type indicates the importance of their interaction in describing control strategy. Individual differences appear the stronger of the factors, halving the number of errors found in a generalization based on encounter variables alone. Pilots appear to develop individualized strategies which are influenced in similar ways by the type of display being used. Individual differences in the adaptation of control strategy to display, however, appear necessary to account for pilot behavior in detail.

IDENTIFICATION OF STRATEGIES

Examining the effects of non-encounter variables by comparing the performance of complete rule sets relies on the Aq algorithm's capability of finding a set of rules embodying whatever consistencies are present in the data. In this usage, ability to phrase noncontradictory rules is more crucial than their generality. When used to identify strategies, however, the generality and performance of particular rules or families of rules becomes of primary importance.

General strategies tend to be somewhat broader than absolute noncontradiction requires. Particular pilots, displays, and encounters often demonstrate slight variations on more basic strategies. In extracting strategies from a rule set it is necessary to consider a number of explicit trade-offs:

1. Discrimination- The strategy should make few false matches

2. Generality- The strategy should apply to many of the examples

3. Uniqueness- Multiple identified strategies should not match the same examples

4. Coverage- Selected strategies should cover a substantial portion of the examples

5. Parsimony- Only a small number of strategies should be identified

In spatial terms these criteria call for partitioning a large part of the attribute space (coverage) into a small number (parsimony) of large (generality), homogeneous (discrimination), non-intersecting (uniqueness) regions. These goals are often conflicting. As the size of regions (and concomitantly coverage of the rule set) increases, so does the likelihood of matching negative examples or intersecting neighboring regions. Identification of strategies requires selection of a subset of representative rules which "best" meet these criteria.

While the appropriate quantification of these criteria is not apparent it is not necessary for a rough identification of major strategies. Selection of a subset of rules from major regions of homogeneity requires only that the analyst simultaneously consider rule performance and region. Once selected, the performance of the reduced rule set can be evaluated and its usefulness as an abstraction of major consistencies in the observations appraised. Other possible selections do not invalidate this choice but merely vary the fineness of detail in exchanging generality for discrimination or parsimony for coverage. The resulting rule sets provide production system models of the observed behaviors. Conditions under which consistent responding failed to occur can be identified as well.

RULE TREES

Since rules may be refinements of one another or otherwise share observations it is necessary to consider rule sets in a way making their redundancy explicit. This is facilitated by representing rules in trees in which successors are subsets of their predecessors. Rules below a selected rule then describe subregions of that rule while the rule, itself, demarcates a subregion of the rules above it. Rules which are not subsets of any other rule form roots.

Well developed tree structures are typical of major strategies. Roots are found in generalizations collapsed across non-encounter variables while more specific generalizations provide refinements and variations on this basic theme attributable to particular pilots, displays, and encounters. Solitary roots by contrast tend to delimit smaller, less populous regions of the attribute space.

Rules from all generalizations, with tau > .01-.02 to exclude those covering only two or three events, were assembled into rule sets for each control action. Trees were then generated for each response type. A rule was considered a subset if:

1. proper subset- its conditions were a subset of its predecessor's

2. phenotypic subset- all events covered were also covered by its predecessor

3. intersecting subset- 90% of events covered were also covered by its predecessor

While representing rules within trees clusters those most closely associated, even roots may share substantial numbers of observations in common. In selecting rules depicting general strategies it is also necessary to consider the uniqueness of these rules which are not quite so closely related. This overlap can be conveniently expressed in an intersection matrix whose entries are the number of common observations for the rules appearing in its indices. Although less complete in its depiction than the rule tree which represents relations directly in the attribute space, the intersection matrix provides a convenient means for representing more isolated regions. In identifying major strategies the analyst may use the structural information provided by rule trees along with the more complete picture of intersections supplied by the matrix to choose rules from among branches, between trees, and among roots. This task will generally prove less formidable than it sounds since a major strategy will usually spawn a tree with a good representative(s) near its root while isolated roots typically have low coverage and may be disregarded.

RESULTS

A set of 9 rules were selected from the generalizations based on rule trees and intersection matrices. The selected rule set covers 44% of the sampled event space with 143 correct matches and 24 errors yielding PRE=.77 and tau=.61. When performance is considered relative to the entire event space these figures become: correct matches=213 errors=171 with PRE=.32 and tau=.27.

Two rules describe conditions for taking no action, three for turning vertically away, and four for turning horizontally toward. Turning vertically toward the intruder occurred very rarely (12 out of 384 encounters) and so was not modeled. The horizontal away response accounting for 70 of the 384 encounters also was not represented. Although 73% of these occurrences are successfully described by a set of 29 horizontal-away rules with only 21 errors, these rules have uniformally small coverage and low overlap. Over half of the horizontal-away rules were restricted to groups of five or fewer pilots indicating the idiosyncratic (or coincidental) nature of this response choice. The overall inconsistency in the choice of this response is revealed in the rule trees for horizontal-away where 27 of 29 rules stand alone if a 90% inclusion criterion is applied. To consider pilot strategy it is necessary to examine the rules, themselves, in greater detail. This will be done for each response class.

NO ACTION

Rule No. 1
[Pilot=7,8,9,10,11,12,13,14,15,16] & [Horizontal passing position=intercepting or in front]
& [Projected horizontal separation=far]
correct matches=22, errors=7, PRE=.09, tau-b=.07

Rule No. 2
[Pilot=3,4,7,8,11,14,15,16] & [Projected horizontal separation=far] &
[Vertical crossover=true]
correct matches=18, errors=6, PRE=.07, tau-b=.05

RULE SET SUMMARY FOR NO-ACTION
correct matches=35, errors=11, PRE=.14, tau-b=.11

The essential condition for eliciting no response appears to be a large projected horizontal separation. Thirty of the 36 rules found for no-response were refinements of this condition, [Projected horizontal separation=far]. Standing alone this condition produces 43 correct matches with 52 errors. The two rules selected miss 8 of these matches but result in 41 fewer errors.

Both individual differences and other aspects of the encounters appear responsible for the increase in selectivity. In the first rule, encounters in which the intruder would pass behind are excluded. This finding is consistent with (O'Connor et.al., 1980) and findings in relation to the horizontal-toward response in this study, that pilots tend to maneuver in a way to cause intruders to pass in front. Individual differences have an equally clear influence. Pilots 7,8,11,14,15,and 16 appear in both of the selected rules. If [Projected horizontal separation=far] is constrained to this group of pilots, correct matches are reduced by only 44% while errors decline by 79%. If the selected rules were restricted to these pilots, selectivity again improves with correct matches declining from 35 to 22 and errors from 11 to 5.

While pilots made no response on only 48 out of the 384 encounters, patterns are found for this choice. A relatively small group (6-8) of pilots account for almost all occasions on which a constant course was maintained. This choice was made appropriately for large horizontal separations but failed to occur when the major separation was vertical or the intruder was oriented to pass behind.

VERTICAL AWAY

Rule No. 3
[Pilot=2,6,7,8,10,11,12,15] & [Projected vertical separation=near] &
[Vertical velocity=zero]
correct matches=43, errors=3, tau-b=.09

Rule No. 4
[Pilot=6,7,11,13,16] & [Projected horizontal separation=near] &
[Projected vertical separation=very near or near] &
[Vertical velocity=zero]
correct matches=27, errors=1, tau-b=.06

Rule No. 5
[Display=no predictor] & [Vertical velocity=zero]
correct matches=27, errors=5, tau-b=.05

RULE SET SUMMARY FOR VERTICAL-AWAY
correct matches=62, errors=7, tau-b=.12

Rules for the vertical away response are contained within the portion of the space in which the intruder is approaching at a constant altitude. Seventy-six of 134 encounters responded to with a vertical-away response were of this type. While 34 additional encounters are covered by 13 more rules in which this condition is not explicitly expressed, their observations fall largely within the constant altitude region. The failure to find strong rules covering the 54 encounters in which the intruder changed altitude indicates an inconsistent usage of the vertical-away response under these conditions. Rules 3 and 4 contain proximity conditions and apply to all displays. In rule 5 both proximity and individual differences are dropped. In the absence of predicted separation pilots chose a vertical away response when confronted with an intruder at constant velocity regardless of the actual threat. Examination of these rules indicates that, for these encounters, projected proximity information influenced the decision to respond but not the response chosen. Pilots' choice of the vertical-away response appears limited to the constant altitude intruder although a strategy of increasing vertical separation would apply to vertically moving intruders as well. The presence of predicted altitudes does not appear to influence this decision. While the vertical-away response was the modal response in this study its association with a clearly discriminable form of separation information rather than projected separations provided by the predictor displays may indicate some difficulties in the use of this information to guide control actions.

HORIZONTAL TOWARD

Rule No. 6
[Pilot=7,11,12,14] & [Display=no predictor or predictor] &
[Projected horizontal separation=very near or near] &
[Vertical crossover=no] & [Vertical velocity=not zero]
correct matches=14, errors=0, PRE=.06, tau-b=.03

Rule No. 7
[Pilot=6,11,12,14] & [Projected horizontal separation=near] &
[Projected vertical separation=far] & [Vertical velocity=not zero]
correct matches=9, errors=0, PRE=.04, tau-b=.02

Rule No. 8
[Pilot=4,5,11,12,15] & [Projected horizontal separation=very near] &
[Vertical velocity=not zero]
correct matches=19, errors=6, PRE=.07, tau-b=.03

Rule No. 9
[Pilot NE 5,6,10] & [Display=no predictor] & [Passing position=intercept or in front]
& [Projected horizontal separation=near] & [Vertical velocity=not zero]
correct matches=15, errors=0, PRE=.06, tau-b=.04

RULE SET SUMMARY FOR HORIZONTAL-TOWARD
correct matches=46, errors=6, PRE=.18, tau-b=.10

Rules for the horizontal-toward response are contained within the complementary "changing intruder altitude" portion of the event space. This factor rather than proximity or relative orientation appears crucial in the choice between horizontal and vertical responding. While the vertical-away response was chosen consistently throughout the constant altitude region, the choice of the horizontal-toward response is less monolithic. As noted in the discussion of "no-response", this choice occurs almost exclusively (98%) in this region. Similarly 83% of the vertical-toward, 43% of the vertical-away, and 93% of the horizontal-away responses occur in encounters in which the intruder is changing altitude.

Rules 6, 7, and 8 are refinements based on individual differences of a strategy of turning toward intruders who are laterally close and changing altitudes:

[Projected horizontal separation=very near or near] &
[Vertical velocity=not zero]
correct matches=85, errors=107, PRE=.12, tau-b=.05

The poor performance of the rule expressed in this general way indicates this strategy is followed by only a small group of pilots. The improved selectivity of rules 6 and 7 is attributable primarily to pilots 11, 12, and 14. The general rule restricted to these pilots:

[Pilot=11,12,14] & [Projected horizontal separation=very near or near]
& [Vertical velocity=not zero]
correct matches=28, errors=8, PRE=.08, tau-b=.04

accounts for 75% of the encounters covered by rules 6 and 7 and represents a major improvement in selectivity over its unrestricted form. Rule 8, another refinement of the general strategy which restricts the rule to intruders at the closest proximity, is followed by a larger group of pilots. None of these rules shares as many as 60% of its observations with the rule embodying the recommended strategy for a horizontal-toward response:

1. There is a threat
2. Maneuvering horizontally toward the intruder will maintain the aircrafts' relative horizontal positions and increase horizontal separation at point of closest approach.

[Passing position=intercept or in front] &
& [Projected horizontal separation=very near or near]
correct matches=65, errors=79, PRE=.02, tau-b=.03

Rule 9, by contrast, is a refinement of the recommended strategy fitting most pilots using the display without a predictor.

DISCUSSION

Within the range of encounters examined, the vertical movement of the intruder appears the most crucial factor in determining the pilot's dominant response. Under conditions in which the intruder approached at a constant altitude pilots under all displays, with few individual differences, and with little regard to the degree of threat, maneuvered vertically away. This strategy follows the principle of least effort in limiting the decision to a single dimension (vertical velocity) and producing a response which increases separation at point of closest approach under all conditions. While ensuring success at the pilots' primary task of avoidance, this strategy may run counter to the secondary task of maintaining course in the face of nonthreatening encounters. This shortcoming is highlighted by noting that of 48 occasions on which the pilot did not maneuver only one occurred under these conditions.

When the intruder was changing altitude the vertical response dimension was largely ignored accounting for the dominant response on only 24% of such occasions. As in previous studies (Palmer et al. 1981, Ellis and Palmer 1982, Smith et al. 1982,1984) horizontal-toward were preferred to horizontal-away responses. Palmer et al. 1981 have attributed this tendency to the pilots' desire to maintain visual contact with the intruder while Ellis and Palmer (1982) have suggested they desire, instead, to minimize the time to resolution of the conflict by passing behind the intruder. Regardless of the motivation, this effect is found consistently in CDTI studies and should be considered in assessing the usefulness of such displays. Smith et al. shed additional light on this preference, finding that encounters rated as less threatening showed a stronger turning-toward tendency. Rules identified for the horizontal-toward response support this view showing a general preference for the horizontal-toward response while using predictor displays which allowed a clear view of conflict resolution but limiting the response to the more conservative recommended strategy when the display lacked predictors.

As found in earlier studies (Smith et al., Palmer et al., Ellis and Palmer) large individual differences were noted among pilot's strategies. The most nearly universal decision was the choice of the vertical-away maneuver under conditions in which it unambiguously increased separation.

The rules identified suggest that vertical information may not be presented in the most useful manner. None of the nine selected rules contain any reference to this relation although it contributes as much to achieved separation and collision avoidance as the horizontal dimension. This neglect is further reflected in the pilots' overall preference for horizontal maneuvers. Smith et al. have suggested the preference for horizontal responses may be due to FAA regulations, comfort, safety or fuel conservation but the absence of vertical information from decision rules suggest the bias may more likely be due to the superior display of horizontal traffic information.

The finding that pilots using predictive CDTI displays were more likely to proceed with conflict resolution by turning toward the intruder than following the recommended strategy reinforces concerns aired by Palmer et al. (1981), Lester and Quan (1983) and others that CDTI in some instances may actually make collisions more likely. Pilots, themselves, are not immune to this fear. The October 28, 1984 New York Times observes that, "The Airline Pilots Association has been especially insistent that the devices must ultimately be able to recommend a horizontal right turn or left turn maneuver in addition to a vertical maneuver." Earlier analysis of this data (Palmer 1983) indicates that the noiseless predictor display led to fewer positive CAS advisories and smaller maneuver magnitudes while the predictorless display resulted in smaller achieved separations and less frequent agreement with the recommended strategy. The present investigation suggests that the superiority in performance on the predictor displays results from improvements in execution rather than fundamental shifts in strategy. For one group of pilots, in fact, consistent violation of the recommended strategy was linked with the use of the noiseless predictor display. While the most widely employed strategy observed was the vertical away response to a constant altitude intruder, vertical responses were generally avoided under other conditions. Since projected altitudes at closest point of approach provide information unavailable from rapidly updating data tags, the failure to find a related consistency in pilots' responses suggests some difficulty in abstracting or using this more detailed altitude information as it is presented.

REFERENCES

1. Curry, R.E. Will the ATSD precipitate display-assisted collisions? in H.G. Weiss and R.W. Bush (eds.), The Role of an Airborne Traffic Situation Display in an Evolving ATC Environment (PB-215-714). Cambridge, Ma: MIT, 1972.

2. Ellis, S.R. and Palmer, E.A. Threat perception while viewing single conflicts on a cockpit display of traffic information (Report No. 81341). moffett Field, Ca.: NASA-Ames Research Center, 1982.

3. Hoff, William A., Michalski, Ryszard S., Stepp, Robert E. "INDUCE 3: a program for learning structural descriptions from examples", in draft form, Department of Computer Science, University of Illinois at Urbana-Champaign, 1983.

4. Lester, P.T. and Quan, E.E. "The cockpit display of traffic information and the threat alert and collision avoidance system integration: a review", Proceedings of the Second Symposium on Aviation Psychology ,Columbus,Ohio: Aviation Psychology Laboratory, Ohio State University, 1983.

5. O'Connor, S., Jago, S., Baty, D. and Palmer, E. The effect of viewing time, time to encounter, and practice on perception of aircraft separation on a cockpit display of traffic information (Report No. 81173). Moffett Field, Ca.: NASA-Ames Research Center, 1980.

6. Palmer, Everett "Conflict resolution maneuvers during near miss encounters with cockpit traffic displays",Proceedings of the Human Factors Society-27th Annual Meeting, Santa Monica, Ca.: Human Factors Society ,757-761, 1983.

7. Palmer, E.A., Jago, S.J., Baty, D.L., and O'Conner, S. Perception of horizontal aircraft separation. on a cockpit display of traffic information. Human Factors, 605-620, 22, 1980.

8. Palmer, E., Jago, S., and DuBord, M. "Horizontal conflict maneuvers with a cockpit display of traffic information", Proceedings of the Seventeenth Annual Conference on Manual Control, Los Angeles, Ca.: UCLA, 1981.

9. Smith, J.D., Ellis, S.R., and Lee, E. Avoidance maneuvers selected while viewing cockpit traffic displays (NASA TM-84269). Moffett Field, Ca. 1982.

10. Smith, J.D., Ellis, S.R., and Lee, C.E. "Perceived threat and avoidance maneuvers in response to cockpit traffic displays", Human Factors ,33-48, 26, 1984.

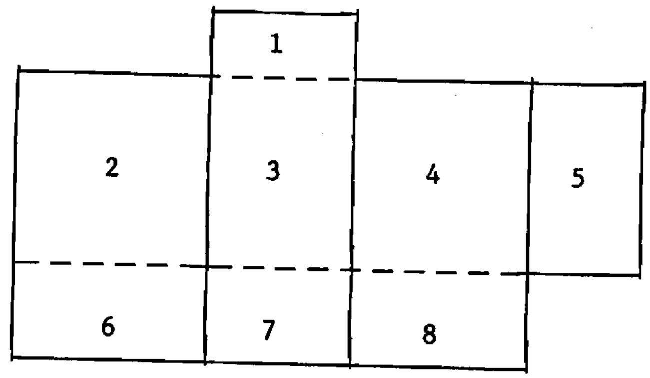


Figure 1

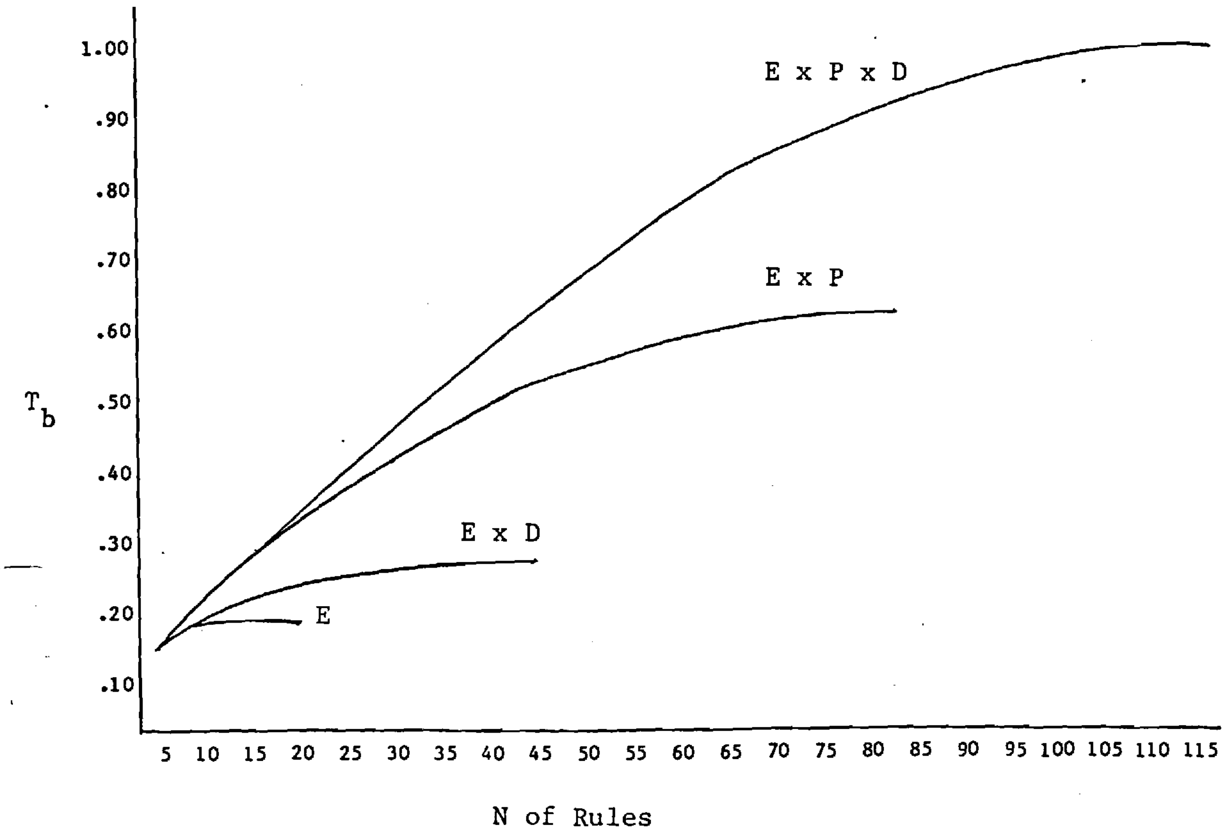


Figure 2

E - encounter

D - display

P - pilot

A MODEL-BASED APPROACH FOR ONLINE AIDING AND TRAINING IN PROCESS CONTROL

Annette Knaeuper
Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

and

Nancy M. Morris
Search Technology, Inc.
Norcross, Georgia 30092

ABSTRACT

This research addressed the feasibility of adapting an existing rule-based system as an online "coach" for controlling PLANT, a simulation of a generic process plant. KARL, a rule-based model capable of controlling PLANT, was adapted to provided three types of information to subjects: 1) situation assessment (i.e., which operational procedure, if any, was applicable for a given situation); 2) guidance in following procedures (i.e., feedback whenever subjects' actions were inconsistent with available procedures); 3) performance feedback (based upon changes in the system's stability). Subjects received this information online while controlling PLANT. Compared to subjects in an earlier experiment who controlled PLANT without the benefit of the coach, these subjects maintained a generally more stable system, scored higher on a paper-and-pencil test of system knowledge, and were more successful in diagnosing an unfamiliar failure of the PLANT safety system. Careful analysis of these results in light of previous research with PLANT indicated that the reasons for these differences were not as straightforward as they might appear. This experiment is viewed as illustrating potential benefits and subtleties of using a rule-based model as an online coach.

INTRODUCTION

As systems increase in complexity, the question of how persons should be trained to operate them becomes more important. The amount of training required for someone to become proficient at controlling a complex system may be quite extensive, and it is necessary to consider a number of issues when developing such a training program. These issues include the content and format of instructional material and the structure of the program. Because of inherent human limitations, it may also be necessary to consider provision of some kind of performance aid, in addition to appropriate training.

Many reports are available which directly or indirectly address issues relevant to training (Morris & Rouse, 1984b). Some are directed at obtaining an understanding of how people solve problems, either in the laboratory or in contact with an actual system. Others investigate the effects of various training approaches upon performance. Often there is a discussion of the human's "mental model" of the system being controlled (Rouse & Morris, 1984).

One study in particular served as a basis for the present research. Morris investigated the effects of different types of instruction upon subjects' ability to control PLANT, a computer-based simulation of a generic fluid production process

(Morris, 1983; Morris & Rouse, 1984a). The PLANT operator's task is to supervise the flow of fluid through a series of tanks interconnected by valves so as to maximize production. This may be done by opening and/or closing valves and adjusting input and output, via commands entered at the terminal keyboard. A number of failures may occur in PLANT, so there are several diagnostic and repair commands available as well.

The primary comparison in Morris' research was between two different types of instruction: 1) operational procedures, and 2) a description of dynamic principles and functional relationships in PLANT. Four groups of subjects were compared, distinguished on the basis of the combination of written instructional materials they received (i.e., principles, procedures, neither principles nor procedures, or both principles and procedures). Instruction was found to have no effect upon subjects' achievement of the overall goal of production, in that there were no differences between groups with respect to this measure. However, those groups receiving procedures were found to control the PLANT in a more stable manner, even though all groups had been told to maintain stability.

An interesting aspect of this research was an investigation of subjects' ability to deal with two unfamiliar failures: a tank rupture, and failure of the PLANT safety system. (The failures were unfamiliar in that, although subjects knew they could occur, they had not experienced them before.) Almost all subjects repaired the tank rupture; however, only half of the subjects in each group successfully diagnosed the safety system failure. This was surprising, because subjects with an understanding of the functioning of the system (as described in the principles) should have been better able to make that diagnosis.

As a result of these findings, it was suggested that one of the reasons a knowledge of principles failed to help many subjects deal with the unfamiliar failure was that those people did not realize that they were in an unusual situation, and thus did not realize that they should use their knowledge. In other words, they failed to make an accurate assessment of the situation. This notion was indirectly supported by the fact that those persons who did repair the unfamiliar safety system failure also maintained a more stable system in general; since the effect of the safety system failure was to make the PLANT appear more unstable, maintaining a stable system may have enabled subjects to detect the presence of an unusual situation more readily.

Some useful insights into subjects' behavior were gained by comparing their performance to that of KARL (Knowledgeable Application of Rule-based Logic), a model capable of controlling PLANT (Knaeuper, 1983; Knaeuper & Rouse, 1984). KARL is a rule-based model patterned after a general model of human problem solving proposed by Rouse (1983), which suggests that problem solving is accomplished in three stages: 1) recognition and classification, 2) planning, and 3) execution and monitoring. These three stages essentially define KARL's structure. When controlling PLANT, KARL accesses a knowledge base consisting basically of information contained in written information available to subjects (i.e., operational heuristics and procedures, and information about dynamic principles and functional relationships).

When the performance of subjects and KARL was compared, it was noted that KARL consistently achieved higher production and maintained a more stable system than did subjects. It was also interesting to examine differences in the courses of action chosen by subjects and KARL in solving problems in PLANT. Basically, two rather systematic differences were found. First, the levels of system input and output chosen by subjects were not as high as those chosen by KARL (and suggested by procedures); subjects were more conservative in this respect. Second, KARL adjusted input and output much more frequently than did subjects; this reflected heuristics within KARL which were directed at maximizing production, which were not a part of operational procedures.

Considering some of the apparent difficulties experienced by subjects in making an accurate situation assessment and following procedures, and the benefits derived from using KARL as an off-line analysis tool, an idea emerged. Why not make it possible for KARL to analyze subjects' actions online and provide advice, thus functioning as an online "coach"? It seemed that such an approach could prove to be useful for both training and aiding.*

DESCRIPTION OF THE COACH

In light of the factors noted above, the decision was made to provide subjects with three types of information. In the context of PLANT, this information was displayed on the terminal near the area where normal operating messages were displayed. The first type of information was related to situation assessment. Specifically, a message informing the subject which procedure was currently applicable was shown (e.g., "Procedure 5"). If no procedure applied, the following message was displayed: "No procedure applicable; Normal tuning".

Subjects also received guidance in following procedures. KARL monitored subjects' actions, and provided feedback if a given action was inconsistent with the applicable procedure. For

---

* Of course, one could view this approach as simply a special case of "expert systems". This issue is discussed later in this paper.

example, the following message might appear: "Your action (cva,e)* is inconsistent with Procedure 5. Keep all valves open until the system is stable again. Type 'y' for change." As may be ascertained from the last portion of the message, subjects had the option of overriding KARL or changing their actions to be consistent with KARL's recommendations.

The third type of information supplied by KARL was <u>performance feedback</u>, or information about the degree to which subjects' actions were succeeding in remedying problems in the system. This information was supplied because of the length of time required for the consequences of actions to become manifest. These messages were based upon changes in PLANT stability over a period of 10 time units, and consisted of the following: "Instability extreme", "Instability excessive", or "Instability improving".

The process of enabling KARL to supply such messages was relatively straightforward. However, when an attempt was made to control PLANT with KARL as an assistant, a number of problems became apparent. For example, KARL's advice as to what actions should be taken was not always consistent with procedures. This could be attributed to the nature of KARL's approach to PLANT.

---

* cva,e = close the valve between tanks a and e

Although the information in the procedures was contained in KARL's knowledge base, KARL also employed several heuristics when controlling PLANT, which occasionally preempted the action recommended in procedures.

Another problem was related to KARL's situation assessment. During the course of PLANT operation, situations would occasionally arise which were "borderline" conditions with respect to the applicability of various procedures. KARL's decisions as to which procedure applied were based upon fixed values of state variables. In borderline situations, normal fluctuations of these state variables caused KARL to change the situation assessment message rather frequently (e.g., every other time unit).

A third source of difficulty was KARL's "persistence" in reporting actions which were inconsistent with procedures. The PLANT operator was given the option of overriding KARL and implementing an action against KARL's recommendations. However, the consequence of thus failing to conform was to receive another message. KARL did not know how to concede; in short, KARL was a nag.

These problems were remedied in two general ways. First, it was necessary to inhibit the display of all messages which were not procedure-oriented. Second, thresholds for prompts were incorporated. For example, if a subject failed to comply with one of KARL's suggestions, KARL did not make the same suggestion again for five time units. As another example, "hysteresis" was introduced into the situation assessment thresholds to avoid the aforementioned problem of borderline conditions.

An experiment was conducted to evaluate the effectiveness of KARL as an assistant. Two general issues were of interest: 1) the feasibility of adapting a rule-based system (which was not originally designed as an aid) to support human problem solving, and 2) the effects of an online coach upon humans' performance.

METHOD

Subjects

Junior and senior undergraduates at Georgia Institute of Technology served as paid volunteer subjects. All eight of them were majors in industrial and systems engineering, and had completed courses in physics, dynamics, calculus, and differential equations.

Experimental Procedure

The experimental procedure in this experiment was almost identical to that used in the research described earlier (Morris, 1983; Morris & Rouse, 1984a). Training provided to subjects in this experiment was equal to the group receiving instruction in both principles and procedures in the earlier experiment, with the exception that aiding was available.

Subjects served in a total of 13 sessions each, with the average length of each session being approximately 60 to 75 minutes. Generally, training was accomplished during the first eight sessions, in which subjects read instructional materials and practiced controlling PLANT. A discussion of principles governing PLANT was provided during session 3, and operational procedures were made available for the first time in session 5. KARL was used as an online coach during sessions 5-8, and supplied the three types of aiding information described earlier.

Sessions 9-13 were considered experimental sessions, in that no further instruction was provided by the experimenter, and no questions from subjects were answered. As with the earlier experiment, unfamiliar situations (i.e., a tank rupture and a safety system failure) were introduced in sessions 10 and 12, which were counterbalanced across subjects. The coach did not provide guidance in following procedures during sessions 9-12; subjects received only information related to situation assessment and overall performance feedback. No information from the coach was available during session 13. At the end of session 13, subjects completed a paper-and-pencil test of knowledge about PLANT and the coach, based upon material contained in the written instructions.

RESULTS

In order to assess the effects of aiding, the performance of subjects in this experiment was compared via analysis of variance to performance of the group receiving both principles and procedures in the earlier PLANT research. (In the following presentation, these groups are referred to as the aided and unaided group, respectively.) Thus, performance measures were used as dependent variables in two-way analyses with one between-subjects factor (aiding) and one within-subjects factor (session).

As with the earlier research, the experimental manipulation had no significant effect upon total production achieved, although the mean for the aided group was slightly higher (344.6 vs. 320.2 units of production per time unit). There was also no significant effect of aiding on the number of automatic valve trips experienced (an indication of PLANT stability). However, as with total production, the mean for the aided group was slightly better (i.e., lower) (0.497 vs. 0.605 trips per time unit).

Aiding also failed to have a significant effect upon another measure of PLANT stability: variance of fluid levels in the system. Once again, the trend was in the expected direction, in that the mean for the aided group was lower (12.44 vs. 15.27).

Two performance measures were significantly affected by aiding. Aided subjects kept a higher percentage of valves open (92% vs. 87%, $p < .04$), and generally maintained a higher level of input into the system (116.8 vs. 106.9 units per time unit, $p < .04$). The practical significance of these results is presented later.

Assessing subjects' performance during unfamiliar situations, there was no effect of aiding upon subjects' repair of the tank rupture (15 of the 16 subjects did so). However, it was found that seven out of eight subjects in the aided group repaired the unfamiliar failure of the PLANT safety system, whereas only three of the eight unaided subjects found that failure. This difference in proportions was found to be statistically significant ($p < .04$).

Differences in scores on the test of PLANT knowledge were examined. Although overall scores did not differ significantly, it was found that the aided group scored significantly higher on the section of the test related to dynamic principles (83% vs. 69%, $p < .05$).

Finally, the actions selected by subjects were compared to actions which would have been selected by KARL in the same situation. This comparison was similar to that reported for the earlier experiment (Knaeuper, 1983; Knaeuper & Rouse, 1984).

There was no significant difference in the degree to which actions chosen by aided and unaided subjects agreed with those selected by KARL.

DISCUSSION AND CONCLUSIONS

As noted in the introduction, this research was prompted by two issues: 1) the feasibility of adapting a rule-based model as an online coach, and 2) the effects of such asistance upon subjects' ability to control PLANT. With regard to the second issue, none of the statistically significant effects the coach had upon subjects' performance were related to primary performance measures. Although mean performance for the aided group was better with all measures, the only significant effects of aiding were upon the secondary performance measures of number of open valves and level of system input. These measures indicate that subjects did what they were told to do. Although all subjects (in this research and in the earlier experiment) were instructed to keep all valves open and maintain a relatively high level of input and output, apparently the coach's presence caused them to follow these instructions more closely.

Whereas it is fairly easy to provide an explanation for subjects' following instructions more closely, explaining why more subjects in the aided group were able to diagnose the safety system failure is not as straightforward. Three possibilities are suggested by the data. First, since failure of the safety system resulted in automatic closing of valves at random, the ability to maintain more valves open in general may have assisted subjects in detecting the presence of an unusual situation. Once detected, it should have been easy to determine that the cause of the unusual situation was failure of the safety system, since only two unusual failures were possible.

Judging from the available evidence, however, it is difficult to imagine that this is a sufficient account of what happened. A look at the performance of all subjects supplied with procedures in the earlier experiment conducted by Morris (i.e., those with procedures only, and those with both procedures and principles) reveals that there was no difference in the number of valves kept open by persons who repaired the safety system and those who did not (89% vs. 88%). Additionally, a subsequent examination of logs kept by the unaided group during the time the safety system had failed indicated that at least six of the eight people felt that something was wrong; yet, only three of these successfully diagnosed the failure, and the others attributed the problem to deficiencies in their control actions.

Another possible explanation may be found in the fact that the aided group scored significantly higher on the test of information related to dynamic principles. Perhaps an increased knowledge of the functioning of the system enabled the aided group to diagnose the unfamiliar failure. This explanation also seems inadequate. There was no difference in the test scores of unaided subjects who repaired the safety system and those who did not (69.3% vs. 69.2%).

The third explanation for aided subjects' success in diagnosing the failure of the safety system is that somehow providing them with the coach made the difference. During the session in which the safety system failed, two types of aiding messages were provided: situation assessment and performance feedback. The situation assessment consisted of informing subjects which procedure, if any, applied. There were no messages such as "unfamiliar situation". Performance feedback was related to changes in the stability of the system. When the safety system failed, it is possible that subjects received conflicting messages, such as "No procedure applicable" and "Instability extreme". Apparent conflict such as this may have served as a cue that something was wrong, and could have suggested to subjects that the problems in the system were not the result of poor control actions.

These ideas about the role of the coach in the unfamiliar situation are purely conjecture at this point. It seems likely that a combination of all of these factors (i.e., increased system stability, knowledge of the functioning of the system, and assistance in situation assessment) contributed to subjects' success. An understanding of factors affecting the human's ability to deal with an unfamiliar event could have important theoretical and practical implications, and further investigation of this issue is warranted.

Finally, another question arises with regard to the results of this research: Why did aided subjects score higher on the test of dynamic principles? Since the primary difference in the way the two groups were treated was the presence or absence of the online coach, it would appear that this was the reason for the difference in the test scores. This is counterintuitive, however, because the focus of the aiding was on following procedures, and not on understanding the functioning of the system. Therefore, interpretation of this result must be delayed until the research can be replicated, using a larger number of subjects and controlling for potential differences in abilities.

Considering the feasibility of adapting a rule-based model as an online coach, this research has served to emphasize the complexities and subtleties of model-based online aiding and training. As noted by other researchers (Clancey & Lestinger, 1982; Jackson & Lefrere, 1984), answering the questions of what advice and feedback to provide, as well as when they should be provided, is far from straightforward. This point is particularly supported by the results reported here where subjects benefited along several dimensions by having an online coach, but did not become more like the coach in the process (i.e., there was no increase in agreement between the subjects' and model's choices of actions). Thus, the results of being coached can be more than, or at least other than, simply gaining the coach's expertise. This has profound implications for the current view of "expert systems" as a panacea for training and aiding.

ACKNOWLEDGEMENTS

This research was partially supported by the Office of Naval Research under Work Unit 154-491 (Contract N00014-82-K-0487), and partially supported by the National Aeronautics and Space Administration under Ames Grant NAG 2-123.

REFERENCES

Clansey, W. J., & Letsinger, R. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching (Tech. Rept. STAN-CS-82-908). Palo Alto, CA: Stanford University, 1982.

Jackson, P., & Lefrere, P. On the application of rule-based techniques to the design of advice-giving systems. International Journal of Man-Machine Studies, 1984, 20, 63-86.

Knaeuper, A. A rule-based model of human problem solving behavior in dynamic environments (Tech. Rept. 83-3). Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, August 1983.

Knaeuper, A., & Rouse, W. B. A rule-based model of human problem-solving behavior in dynamic environments. IEEE Transactions on Systems, Man, and Cybernetics, 1984, in press.

Morris, N. M. Human problem solving in process control (Tech. Rept. 83-2). Atlanta, GA: Center for Man-Machine Systems Research, Georgia Institute of Technology, August 1983.

Morris, N. M., & Rouse, W. B. The effects of type of knowledge upon human problem solving in a process control task. _IEEE Transactions on Systems, Man, and Cybernetics_, 1984, in press. (a)

Morris, N. M., & Rouse, W. B. _Review and evaluation of empirical research in troubleshooting_. Norcross, GA: Search Technology, August 1984. (b)

Rouse, W. B., & Morris, N. M. _On looking into the black box: Prospects and limits in the search for mental models_. Norcross, GA: Search Technology, July 1984.

E-21-

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Reports

August 1984 - February 1985
March 1985 - July 1985

William B. Rouse, Principal Investigator

John M. Hammer, Co-Principal Investigator

Christine M. Mitchell

Nancy M. Morris

C. Michael Lewis

Wan C. Yoon

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

This report covers progress during two six-month reporting periods: 1) August 1984 - February 1984, 2) March 1985 - July 1985. During these periods substantial progress has been made in three areas.

In the rule-based modeling area, Mike Lewis' Ph.D. work is nearing completion. This report includes two papers related to identification and significance testing of rule-based models, and a third paper on an application to CDTI data.

In the area of operator aiding, Wan Yoon's Ph.D. research is focusing on aiding operators in novel failure situations. Chris Mitchell has developed a discrete control modeling approach to aiding PLANT operators. Finally, Bill Rouse and Nancy Morris have developed a set of guidelines for implementing automation.

The third area of progress is the flight simulator hardware and software. While this development effort has taken much more time than originally envisioned, the hardware will be complete within two months and initial simulation software will then be integrated and tested.

SIGNIFICANCE TESTING OF RULE-BASED MODELS

The appended article by Lewis and Hammer describes how to test the significance of rules in rule-based models. The danger in rule-based model building is that the overall model may fit the data well but that individual rules may not contribute to this fit. The article explains several relatively easy methods for testing rule fit.

IDENTIFICATION OF RULE-BASED MODELS

The Human Factors and IEEE SMC conference papers on rule identification included with this report recapitulate much of the work in the re-analysis of CDTI data [Palmer 1983] contained in our last report. Attention over the interim has largely focused on significance testing in the identification process. The SMC paper concentrates on methodological difficulties inherent in employing logical generalization and points out some of the strengths of alternate approaches.

The possibility of developing significance tests for logical generalization remains a paramount advantage over the top-down approaches. The Monte Carlo procedure described in the SMC and Human Factors papers proved effective but inefficient. Running in the background at low priority it has taken about 10 minutes per iteration. One thousand iterations are used. The working paper on representation describes well formed formulas in VL1 which might be used in deriving a closed form significance test. If attainable it would avoid the inefficiencies of repeated search by generating counts of rules directly.

The primary effort in rule identification is now being directed toward the PLANT [Morris 1983] data. Preliminary identification of rules has indicated a lack of stationarity associated with shifts in operator goals and phases of operation. This result is not unexpected as task constraints led KARL [Knaeuper 1983], a production system model of the operator, to employ explicit state->state rules to achieve such transitions.

Since these shifts are unobservable, state vectors will be augmented with an oracle variable encoding shifts KARL "would have made". Shifts in phase dictated by a discrete control model of PLANT [Mitchell 1984] will also be employed in a parallel effort.

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

An aid containing a qualitative device model and designed to counteract human decision-making biases is being investigated. This aid is designed to be used during novel failures, which are defined as failures that are not covered by the operator's training or procedures. The remainder of this section covers the qualitative model, decision aiding, and the applicability of the aid. Further details are in an appended paper.

A qualitative model [deKleer and Brown, Davis, Forbus] represents a physical device as a network of components and connections. Each component and connection can have several discrete states. The behavior of a component (in terms of connection flows such as current) is governed by rules. Component state transition is also governed by transition rules. A solution to a diagnostic problem is an assignment of states to components and connections that explains the observed symptoms and obeys physics as described by the rules. The qualitative model is included in the aid to assist the human in reasoning about the physical device.

The aid also is designed to counteract some human decision making biases. This aiding takes a number of forms. Working memory is augmented with the display of hypotheses and data. The human tendency to forget or overlook is counteracted by the aid's mechanistic reasoning.

Applicability

The applicability of the kind of aid is whenever a novel failure occurs. In commercial aviation, such failures are relatively rare. They would seem to be more common in process control and space flight. The most applicable area would seem to be commercial space loads. Because most of these are one shot, non-life threatening tasks, the operator's training will be limited than on the operation of spacecraft itself. The economic consequences are high for an improperly diagnosed payload problem.

A DISCRETE CONTROL MODEL OF PLANT

The appended working paper by Chris Mitchell develops a discrete control model of the PLANT process control simulation and discusses the potential use of the model as a basis for a new human-computer interface for PLANT.

GUIDELINES FOR IMPLEMENTING AUTOMATION

The appended paper by Bill Rouse and Nancy Morris summarizes recent efforts to understand how people perceive automation and the influence of these perceptions on acceptance of automation. A set of eight guidelines are proposed as a possible means of enhancing acceptance.

FLIGHT SIMULATOR HARDWARE AND SOFTWARE

The hardware is scheduled for completion by September 15, 1985. The following have been completed.

1. Wiring to the sensors
2. Pedestal has been reinstalled.
3. Keyboards are mounted in the pedestal.
4. The CRT's have all been tested and mounted.
5. A force feel system with trim for the elevators was designed and installed.

Progress on the software is as follows.

1. An engine display program has been completed.
2. A very simple flight instrument display has been completed.
3. The existing simulation has been revised to run under UNIX.

The hardware changes that remain are:

1. Cooling fans must be installed for the CRT's.
2. The glare shield must be installed.
3. Some metal panels must be added to shroud CRT's and otherwise close up the cockpit.

The software that remains to be done can be categorized as follows.

1. Modifying the simulation to accept inputs from the A/D converter.
2. Modifying the simulation program to drive different, multiple displays for the instruments.
3. Integration - simply making sure that everything is connected and works the way it is supposed to.

The completion of the above will demonstrate that the simulator will work.

The simulator has come along at a much slower rate than anticipated. We did not initially realize the complexity of the project. As compensation, the final cost of the simulator will be a small fraction of the cost of a new one.

REFERENCES

1. Davis, R. "Reasoning from first principles in electronic troubleshooting," Int'l. J. Man-Machine Studies, V19, 1983.

2. deKleer, J. and Brown, J.S., "Assumptions and ambiguities in mechanistic mental models," in Gentner and Stevens (eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum, 1983.

3. Forbus, K.D., "Qualitative reasoning about space and motion," in Gentner and Stevens (eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum, 1983.

4. Knaeuper, Annette. A model of human problem solving in dynamic environments. M.S. Thesis. Georgia Institute of Technology, 1983.

5. Michalski, R.S. Knowledge repair mechanisms: evolution vs. revolution. Proceedings of the Third International Machine Learning Workshop. New Brunswick, N.J.: Rutgers University, 116-119, 1985a.

6. Michalski, R.S. and Winston, P.H. Variable Precision Logic, A.I. Memo. Boston, Mass.: Artificial Intelligence Laboratory, M.I.T., 1985b.

7. Michalski, R.S. and Reinke, R. Incremental Learning of Concept Descriptions, Machine Intelligence 11, J.E. Hayes, D. Michie, and J. Richards, eds., Oxford: Oxford University Press, 1985c.

8. Mitchell, Christine M. A discrete control model of PLANT, working paper. Center for Man-Machine Systems Research, Georgia Institute of Technology, 1984.

9. Morris, Nancy M. Human problem solving in process control, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1983.

10. Palmer, Everett. "Conflict resolution maneuvers during near miss encounters with cockpit traffic displays", Proceedings of the Human Factors Society-27th Annual Meeting, Santa Monica, CA.: Human Factors Society, 757-761, 1983.

APPENDIX

RULE-BASED ANALYSIS OF PILOT DECISIONS

C. Michael Lewis
Center for Man-Macnine Systems Research
School of Industrial and Systems Engineering
and
School of Psychology
Georgia Institute of Technology

This paper will appear in the Proceedings of the Human Factors Society, 29th Annual Meeting, Baltimore, Md., October, 1985.

ABSTRACT

Rule identification is proposed as an alternative to parameter estimation in the analysis of human performance data. The relation between the choice of language and identifiable consistencies is discussed. Advantages of production system models for the description of complex human behaviors are examined. Threats to validity posed by the use of flexible languages in data analysis are examined. Contrivedness, defined by Eilbert and Christensen (1982) as, "..the tendency of a search procedure to uncover apparent patterns where none exist", is advanced as the major inferential hazard to rule identification. A Monte Carlo significance testing procedure to deal with this threat is proposed. Nonparametric measures of relation, tau-b and PRE, are presented as appropriate performance measures for rule-based models. A rule-based analysis of data from an experiment (Palmer, 1983) involving pilot decisions in air traffic conflicts is presented. Identified rules indicate satisficing with respect to the secondary goal of maintaining a constant course. Reduction in dimensionality of utilized information led to errors for a subgroup of pilots.

INTRODUCTION

Suppose an experiment were conducted in a cooking class. Students have been observed preparing a hollandaise sauce. Measurements on interval scales are available of ingredients, cooking times, and temperatures. A reliable gourmet has rated each student's sauce. Could a regression equation or discriminant function provide adequate information to judge a future hollandaise? No cook would consider compensating for a shortage of eggs by adding extra salt yet this is exactly the remedy such models would suggest. An expression which specified allowable ranges for ingredients, temperatures, and cooking times would fare better.

Such data are fairly common. Crystal formation, catalytic reactions, and disease diagnosis all appear to depend more on the synthesis of values than the sum of their individual effects. These types of relations are frequently encountered in human behavior. In problem solving a subject commonly has a variety of available actions. His task lies in choosing a sequence consonant with some goal. The salience of variables will often depend on their context. For example, functioning headlights may be a useful cue in diagnosing an electrical fault yet be irrelevant to fixing a flat tire. This dependence on context and qualitative differences among responses are hallmarks of data in which the coincidence of values dominates their individual contributions.

Language and Description

Formal description of regularities in data hinges on the choice of language. The validity of generalizations depends on the ability of the chosen language to describe the consistencies present in the data. Languages consist of primitive elements such as attributes, connectives, and operators and a syntax which prescribes the ways in which they may be combined to obtain meaningful statements. The general linear model (GLM) which includes multiple regression, ANOVA, and discriminant analysis is the language of choice in the description of human performance. It has strong advantages in statistical inference and a ubiquity of description, yet the rigidity of its syntax renders it incapable of expressing many of the observable consistencies in human behavior. In essence GLM allows but a single statement:

$$Y = B0 + B1x1 + .. B12x1x2 .. + e$$

While well suited to describing data which fits this mold (e.g., stimulus intensity and reaction time), it becomes nonsensical when turned toward domains such as problem solving or process control. In these more complex domains studied in man-machine interactions, an adequate description must capture both the complexity of the environment and the heterogeneity of responses within a common representation.

Production Systems

A representation gaining currency for describing complex human behavior is the production system. Production systems models have enjoyed success in domains as varied as: crypto arithmetic, logic, and games (Newell and Simon, 1972), air traffic control (Wesson, 1977), fault diagnosis (Hunt and Rouse, 1984), and others. In its simplest form a production system is a set of statements assigning values. Production rules consist of two parts: the condition and the action. When values in the domain of the production system satisfy the conditions of a rule, the rule is "fired" and its action taken.

This ability to accommodate assignments (responses) of arbitrary form makes production systems particularly attractive descriptions for complex behaviors.

Rule Identification

Unfortunately there has been no purely objective way to identify rules from data. As in the pioneering problem solving work of Newell and Simon (1972), analysts have been forced to rely on their own powers of pattern recognition to frame rules consistent with their data.

Previous work identifying rules from data has largely been restricted to expert systems. The most venerable of these efforts, METADENDRAL (Mitchell, 1977), supplied cleavage rules to DENDRAL, an expert system in the area of mass spectroscopy. Michalski (1980) employed GEM, a program based on the Aq algorithm (used in this study), to find rules for the diagnosis of soybean diseases that outperformed an expert system built in the customary manner.

Such efforts, however, have employed rule identification as an adjunct to subjective identification and placed primary emphasis on their models rather than their data. As a technique for data analysis, rule identification has remained largely unexploited. In the present study the propositional logic module (Aq algorithm) of a machine learning program, INDUCE 3 (Michalski, 1982), was used to identify rules embodying consistencies among conditions defining aircraft encounters and corresponding pilot maneuvers.

The Aq algorithm identifies rules in the VL1 language, a simplified propositional logic. VL1 contains only single place predicates and allows only one predicate per class of predicates to be true in a particular statement. Disjunction is only allowed within classes of predicates. For example, if color of bird is a predicate class, then black(bird) V blue(bird) is an allowable phrase. Conjunction is only allowed between classes of predicates. For example, if type of bird is another predicate, then black(bird) & is-raven(bird) is an allowable phrase.

For notational convenience, Michalski (1982) defines a new entity, a selector, to describe classes of predicates. A selector names the predicate class and lists the predicates involved in a disjunction.

Example: [color-bird=black V blue].

Redefined in terms of selectors, VL1 allows only conjunctions of selectors. Since generalization on the preconditions is performed to discriminate among actions, the relationship between precondition and action is represented as implication, ->. An example of a statement in VL1 is:

[x1=*a* V *b*] & [x2=*d* V *e*]-> F

This rule says that if the predicate evaluating true for class x1 is *a* or *b* and class x2 is *d* or *e*, then F(observation) is true. In terms of data consisting of stimuli and responses, VL1 rules identify stimulus conditions under which particular responses have consistently occurred.

Significance Testing

A nonparametric analogue to the coefficient of determination, tau-b (Margolin and Light, 1971), was used to determine the percentage of variance in the actions explained by rules and rule sets. Individual rules and the disjunction of rules issuing a particular action were evaluated in this way.

A similar statistic, PRE (proportional reduction in error) (Bishop et al., 1975), measuring the reduction in error achieved by predicting actions based on the rules rather than assigning the modal action under all rules, is also reported. While inapplicable to rules assigning the modal action, PRE has the advantage of being directly interpretable as the gain in predictability attributable to a rule(s).

Rule identification has been introduced as a process of obtaining consistent generalizations from data. To stop here as machine learning programs do, amounts to little more than fitting a "form gauge" to the data. Inductive inference requires both a consistent description and some measure of confidence that the description embodies actual relations rather than idiosyncrasies of the data/identification method. The syntactic rigidity of GLM reduces this problem to estimating sampling error. Even here, when misapplied as in the case of regressions drawing upon large numbers of variables (Forsythe et al., 1973), opportunism in sampling from a space of descriptions may result in unwarranted inferences.

For logical induction the hazards are reversed. The number of consistent statements which can be made about a set of observations is much greater than the number of observations. As a consequence expressions of confidence should explicitly consider sampling in a space of descriptions. Eilbert and Christensen (1982) refer to this problem as contrivedness, "...the tendency of a search procedure to uncover apparent patterns where none exist."

A logical benchmark to gauge our confidence, is the situation in which there is no relation between the conditions of a rule and the associated response. If this were true, then any pairing of observed conditions and responses would be equally likely. A distribution corresponding to this benchmark can be obtained by considering all permutations of responses with respect to the conditions. The relative frequency of permutations in which rules of equal or greater "quality" are identified provides a measure of the unlikeliness (significance) of identifying rules of the "quality" obtained under a null hypothesis of no relation.

All rules presented in this paper were significant, $p<.01$, using the conventional G-square approximation of chi-square associated with the tau-b statistic. When tested using an experimental 500 permutation Monte Carlo procedure based on controlling contrivedness, only the rules for the vertical-away response were found significant at conventional ($p<.05$) levels. Additional methods of significance testing for rules and rule-based models are discussed in Lewis and Hammer (in press).

CDTI DATA ANALYSIS

Data from an experiment by Palmer (1983) investigating the effects of information quality and intruder characteristics in the use of a cockpit display of traffic information instrument (CDTI) has been reanalyzed using rule identification techniques.

In this experiment sixteen pilots "flew" sixteen programmed encounters under three display conditions. Pilots were instructed to maintain a steady course, using the autopilot unless they received a threat advisory. In response to the threat they were to maneuver to maintain a horizontal separation of greater than 1.5 nm and a vertical separation in excess of 500 ft. They were advised that an appropriate strategy was to maneuver so that the intruder would pass further away but in the same orientation at the point of closest approach.

In the least informative condition, the display portrayed the relative positions of the ownship and the intruder along with tags showing their altitudes. The predictive display provided ground referenced predictors showing predicted positions of the ownship and intruder as well as a tag showing the intruder's projected altitude at time of closest approach. In the third condition, noise was introduced into the predictive display

Explanatory Variables

In this analysis encounter variables, describing the physical relationship between the intruder and ownship which the pilot is instructed to control, were differentiated from experimental variables. Five encounter variables were used. Four describe the relative positions of the aircraft at their point of closest approach as projected at time of alarm. The fifth measure, intruder vertical velocity, remains constant throughout the encounter.

Two non-encounter variables were considered, display type and pilot. Display type in conjunction with the encounter variables describes the stimuli under which a decision is made. Inclusion of pilot identification in the generalization introduces individual differences.

Response Measures

Pilots' responses were represented in terms of maneuvers toward or away from the intruder along a dominant axis. The dominant axis was determined by comparing the ratio of the horizontal and vertical magnitudes of a maneuver to the respective tolerances which the pilots had been instructed to maintain. Five response classes result: 1) no action, 2) vertical-toward, 3) vertical-away, 4) horizontal-toward, and 5) horizontal-away.

Results

A set of 9 rules were selected from the generalizations. The selected rule set covers 44% of the sampled event space with 143 correct matches and 24 errors yielding PRE=.77 and tau-b=.61.

Two rules describe conditions for taking no action, three for turning vertically away, and four for turning horizontally toward. Turning vertically toward the intruder occurred very rarely (12 out of 384 encounters) and so was not modeled. The horizontal away response accounting for 70 of the 384 encounters was also not represented. Although 73% of these occurrences are successfully described by a set of 29 horizontal-away rules with only 21 errors, these rules have uniformly small coverage and low overlap. Over half of the horizontal-away rules were restricted to groups of five or fewer pilots indicating the idiosyncratic (or coincidental) nature of this response choice. The following tables summarize the performance of the selected rule set.

Rule Performance Summary

Act	Hit	Fa	T-b	Pre
NA	22	7	.07	.09
NA	18	6	.05	.07
H-T	14	0	.03	.06
H-T	19	6	.03	.07
H-T	15	0	.04	.06
H-T	9	0	.02	.04
V-A	43	3	.09	
V-A	27	1	.06	
V-A	27	5	.05	

Table 1

Performance Summary by Control Action

Act	Hit	Fa	T-b	Pre
NA	35	11	.11	.14
H-T	46	6	.10	.18
V-A	62	7	.12	

Table 2

Discussion

Within the range of encounters examined, the vertical movement of the intruder appears the most crucial factor in determining the pilot's dominant response. Under conditions in which the intruder approached at a constant altitude, pilots under all displays, with few individual differences, and with little regard to the degree of threat, maneuvered vertically away. This strategy follows the principle of least effort in limiting the decision to a single dimension (vertical velocity) and producing a response which increases separation at point of closest approach under all conditions. While ensuring success at the pilots' primary task of avoidance, this strategy may run counter to the secondary task of maintaining course in the face of nonthreatening encounters. This shortcoming is highlighted by noting that of 48 occasions on which the pilot did not maneuver, only one occurred under these conditions.

When the intruder was changing altitude the vertical response dimension was largely ignored accounting for the dominant response on only 24% of such occasions. As in previous studies (Palmer et al., 1981; Ellis and Palmer, 1982; Smith et al., 1984) horizontal-toward were preferred to horizontal-away responses. Palmer et al. (1981) have attributed this tendency to the pilots' desire to maintain visual contact with the intruder. Ellis and Palmer (1982) have suggested that the pilots desire, instead, to minimize the time to resolution of the conflict by passing behind the intruder. Regardless of the motivation, this effect is found consistently in CDTI studies and should be taken into consideration in assessing the usefulness of such displays. Smith et al. (1984) shed additional light on this preference, finding that encounters rated as less threatening showed a stronger turning-toward tendency. Rules identified for the horizontal-toward response support this view showing a general preference for the horizontal-toward response while using predictor displays which allowed a clear view of conflict resolution but limiting the response to the more conservative recommended strategy when the display lacked predictors.

As found in earlier studies (Smith et al., 1984; Palmer et al., 1981; Ellis and Palmer, 1982), large individual differences were noted among pilots' strategies. The most nearly universal decision was the choice of the vertical-away maneuver under conditions in which it unambiguously increased separation.

The identified rules suggest that vertical information may not be presented in the most useful manner. None of the nine selected rules contain any reference to this relation although it contributes as much to achieved separation and collision avoidance as the horizontal dimension. This neglect is further reflected in the pilots' overall preference for horizontal maneuvers. Smith et al. (1984) have suggested that the preference for horizontal responses may be due to FAA regulations, comfort, safety or fuel conservation. The absence of vertical information from decision rules, however, suggests that the bias may more likely be due to the superior display of horizontal traffic information.

The finding that pilots using predictive CDTI displays were more likely to proceed with conflict resolution by turning toward the intruder than following the recommended strategy reinforces concerns aired by Palmer et al. (1981), Lester and Quan (1983) and others that CDTI in some instances may actually make collisions more likely. Pilots themselves are not immune to this fear. The October 28, 1984 New York Times observes that, "The Airline Pilots Association has been especially insistent that the devices must ultimately be able to recommend a horizontal right turn or left turn maneuver in addition to a vertical maneuver."

Earlier analysis of this data (Palmer, 1983) indicates that the noiseless predictor display led to fewer positive CAS advisories and smaller maneuver magnitudes while the predictorless display resulted in smaller achieved separations and less frequent agreement with the recommended strategy. The present investigation suggests that the superiority in performance on the predictor displays results from improvements in execution rather than fundamental shifts in strategy. For one group of pilots, in fact, consistent violation of the recommended strategy was linked with the use of the noiseless predictor display. While the most widely employed strategy observed was the vertical-away response to a constant altitude intruder, vertical responses were generally avoided under other conditions. Since projected altitudes at closest point of approach provide information unavailable from rapidly updating data tags, the failure to find a related consistency in pilots' responses suggests some difficulty in abstracting or using this more detailed altitude information as it is presented.

A general picture of encounter resolution as a decision sequence emerges from this data (as illustrated in Figure 0-1):

1- If the intruder is approaching at a constant altitude, a vertical-away response is chosen. This decision requires minimal effort since it will always agree with the instructions to increase separation in the same orientation as would have occurred in the absence of a maneuver.

2- A subgroup of pilots chose not to maneuver in non threatening encounters in accordance with their second instruction of maintaining a constant course. This decision comes second since it was restricted to encounters in which the constant altitude condition was not met.

3- A second subgroup of pilots chose a horizontal-toward response for horizontally near intruders in conflict with separation instructions for intruders which would have passed behind.

4- The horizontal-toward decision for pilots as a whole introduced the additional stipulation that the intruder would pass in front bringing the decision into agreement with instructions.

PILOT DECISION SEQUENCE

```
 ||
 \/
 /\   Intruder changing
/  \      altitude
\  / -------------------> Vertical-Away
 \/
 ||
 \/
 /\      Intruder
/  \    threatening
\  / -------------------> No-Action
 \/      (subgroup)
 ||
 \/
 /\  Intruder too close
/  \    horizontally
\  / -------------------> Horizontal-Toward
 \/      (subgroup)
 ||
 \/ Intruder too close
 /\   horizontally &
/  \ would pass behind
\  / -------------------> Horizontal-Toward
 \/
 ||
 \/
```

Figure 0-1

In this sequence we see satisficing by pilots choosing the simplest stratagem (turning away from a constant altitude intruder) when possible before considering the potential threat of the intruder. In the incorrect variant of the horizontal-toward response, a subset of pilots simplified their choice by considering proximity information only, ignoring the more complicated encounter geometry.

SUMMARY

This paper illustrates some of the advantages to be gained through the use of more flexible languages in the analysis of human performance data. The focus of rules on the conditions under which behavior occurs, allows an analyst to pinpoint sources of error directly. The ability of rules to describe behavior itself, rather than derivative measures makes the analysis interpretable in terms of the task rather than abstractions such as effects or interactions. In activities such as reliability analysis, rule identification offers a means for determining the most likely operator actions under conditions of interest. With the emergence of technologies placing logical and algebraic expressions on a more equal footing, the human factors community can be expected to put these new tools to good use.

ACKNOWLEDGMENT

This research is supported by the National Aeronautics and Space Administration under Ames Grant No. NAG 2-123.

REFERENCES

Eilbert, R.F. and Christensen, R.A. "Contrivedness: the boundary between pattern recognition and numerology", Pattern Recognition, 1982, 253-261.

Ellis, S.R. and Palmer, E.A. "Threat perception while viewing single conflicts on a cockpit display of traffic information", (Report No. 81341) Moffett Field, Ca.: NASA-Ames Research Center, 1982.

Forsythe, A.B, Laszlo, J., Jennrich, K., and May, P.R.A. "A Stopping Rule for Variable Selection in Multiple Regression", Journal of the American Statistical Association, 68, 1973, 75-77.

Hunt,R.W. and Rouse,W.B. "A fuzzy rule-based model of human problem solving", IEEE Transactions on Systems, Man, and Cybernetics, 14, 1984, 112-119.

Lester, P.T. and Quan, E.E. "The cockpit display of traffic information and the threat alert and collision avoidance system integration: a review", Proceedings of the Second Symposium on Aviation Psychology, Columbus, Ohio: Aviation Psychology Laboratory, Ohio State University, 1983.

Lewis, C.M. and Hammer, J.M. "Significance testing of rules in rule-based models of human problem solving", to appear in: IEEE Transactions on Systems, Man, and Cybernetics.

Margolin, B.H. and Light, R.J. "An analysis of variance for categorical data II: small sample comparisons with chi-square and other competitors", _Journal of the American Statistical Association_, 69, 1974.

Michalski, Ryszard S. "Pattern recognition as rule guided inference", _IEEE Transactions on Pattern Analysis and Machine Intelligence_, 4, 1982, 349-361.

Michalski, R.S. and Chilausky, R.L. "Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis", _Policy Analysis and Information Systems_, 1980, 125-160.

Mitchell, T. M. "Version spaces: a candidate elimination approach to rule learning", _Proceedings of the Fifth International Joint Conference on Artificial Intelligence_, IJCAI, Cambridge, Mass., 1977, 305-310.

Newell, A. and Simon, H.A., _Human Problem Solving_. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

Palmer, E. "Conflict resolution maneuvers during near miss encounters with cockpit traffic displays", _Proceedings of the Human Factors Society-27th Annual Meeting_, Santa Monica, Ca.: Human Factors Society, 757-761, 1983.

Palmer, E., Jago, S., and DuBord, M. "Horizontal conflict maneuvers with a cockpit display of traffic information", _Proceedings of the Seventeenth Annual Conference on Manual Control_, Los Angeles, Ca.: UCLA, 1981.

Smith, J.D., Ellis, S.R., and Lee, C.E. "Perceived threat and avoidance maneuvers in response to cockpit traffic displays", _Human Factors_, Santa Monica, Ca.: Human Factors Society, 33-48, 26, 1984.

Wesson, R.B. "Planning in the world of the air traffic controller", _Proceedings of the 5th International Joint Conference on Artificial Intelligence_, M.I.T., Cambridge, Mass., 1977, 473-479.

ISSUES IN RULE IDENTIFICATION AND LOGICAL INDUCTION

C. Michael Lewis
Center for Man-Machine Systems Research
School of Industrial and Systems Engineering
and
School of Psychology
Georgia Institute of Technology

This paper will appear in the Proceedings of the 1985 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, Az., November, 1985.

Abstract

The relationship between language and empirical fitting of data is discussed. The production system is presented as an appropriate description of human behavior in Man-Machine systems. Issues arising in the identification of rules from data are examined. Rules identified through logical generalization are shown to be equivocal. Difficulties arising from the use of logic-based procedures with human perfomance data containing errors are explored. Problems relating to rule sets which are not disjoint are discussed and a solution presented. Significant testing issues are raised for rule identification and a procedure based on controlling contrivedness is presented. A synthesis of data and knowledge-based approaches is suggested as a remedy to many of the difficulties discussed.

Introduction

Suppose an experiment were conducted in a cooking class. Students have been observed preparing a hollandaise sauce. Measurements on interval scales are available of ingredients, cooking times, and temperatures. A reliable gourmet has rated each student's sauce. Could a regression equation or discriminant function provide adequate information to judge a future hollandaise? No cook would consider compensating for a shortage of eggs by adding extra salt yet this is exactly the remedy such models would suggest. An expression which specified allowable ranges for ingredients, temperatures, and cooking times would fare better. Such data are fairly common. Crystal formation, catalytic reactions, and disease diagnosis all appear to depend more on the synthesis of values than the sum of their individual effects. These types of relations are frequently encountered in human behavior. In problem solving a subject commonly has a variety of available actions. His task lies in choosing a sequence consonant with some goal. The salience of variables will often depend on their context. For example, functioning headlights may be a useful cue in diagnosing an electrical fault yet be irrelevant to fixing a flat tire. This dependence on context and qualitative differences among responses are hallmarks of data in which the coincidence of values dominates their individual contributions.

Language and Description

Formal description of regularities in data hinges on the choice of language. The fidelity of generalizations depends on the ability of the chosen language to describe the consistencies present in the data. Languages consist of primitive elements such as attributes, connectives, and operators and a syntax which prescribes the ways in which they may be combined to obtain meaningful statements.

Characteristics of formal descriptions often go unacknowledged in the study of human performance. To illustrate the strengths and weaknesses of formal descriptions, a mock experiment was simulated in which multiple linear regression was used to describe the behavior of falling bodies. Data with 5 replications per cell were generated for a one-gram ping pong ball and a ten-gram rubber ball dropped from distances of 10, 20, and 30 feet. The ping pong ball was assumed to accelerate at 28 ft./sec**2. Errors in measurement for time were normally distributed with standard deviation = .1 sec. The resulting description was: time (in seconds) = .6 + .03 * distance (in feet).

This apocryphal example illustrates two important points about description languages and equivocality (indeterminancy of identification).

1- The empirical fitting of models to phenomena is equivocal with respect to languages. For this data, the above new "law" fits as well as Newton's law.

2- Simplicity is favored in the choice of a description language. The complexity of a language sufficient to represent Newton's law might yield other descriptions also.

This example is not meant to ridicule empirical applications of linear models to data, but to illuminate the nature of the enterprise. The description is, in fact, excellent by the standards of behavioral research ($R^2 = .84$). The Aristotelian notion that rate of descent is determined by weight was correctly disconfirmed ($p < .076$) and the positive relation between distance and time correctly identified.

The general linear model (GLM) has been the language of choice in the description of human performance. It has strong advantages in statistical inference and ubiquity of description, yet the rigidity of its syntax renders it incapable of expressing many of the observable consistencies in human behavior. While well-suited to describing data that fits this mold (e.g., stimulus intensity and reaction time), GLM becomes nonsensical when turned toward domains such as problem solving or process control. In these more complex domains, studied in man-machine interactions, an adequate description must capture both the qualitative complexity of the environment, and the heterogeneity of responses within a common representation.

<u>Production Systems</u>

A representation gaining currency for describing complex human behavior is the production system. Production systems models have enjoyed success in domains as varied as: crypto arithmetic, logic, and games [8], air traffic control [12], fault diagnosis [3] and others. In its simplest form a production system is a set of statements assigning values. Production rules consist of two parts: the condition and the action. When values in the domain of the production system satisfy the conditions of a rule, this rule is "fired" and its action taken.

This ability to accommodate assignments (responses) of arbitrary form make production systems particularly attractive descriptions for complex behaviors. Over the past two years, our research has been concerned with the identification of production rules from human performance data.

<u>Data</u>

Data from an experiment, reported in [9], dealing with aircraft encounters and data from a process control task, reported in [7], were chosen for this effort. INDUCE [6], a machine learning program, was employed to identify rules that would describe the human performance in these experiments.

In the study reported in [9], pilots using a Cockpit Display of Traffic Information (CDTI) with no predictor, predictor, or predictor with noise were required to maneuver away from an intruder. Each trial consisted of a single maneuver. For purposes of rule identification maneuvers were classified according to their dominant axis of response. The VL1 language, a propositional logic, was used for description.

In the study reported in [7], subjects controlled PLANT, a simulated process plant. Sequenced control actions were treated as replications. Historical variables were included to provide information on PLANT dynamics to induce stationarity. The VL2 language, a predicate logic, was used for description.

Learning Rules

Both weak (domain independent) and strong (domain-based) methods have been applied to the problem of rule learning. Michalski [4] refers to these approaches as revolutionary and evolutionary respectively.

In evolutionary procedures examples are used (usually sequentially) to refine programmer supplied rules. Programs which improve their performance with experience [10] are the most frequent exemplars of this approach. With few exceptions, identification methods applied successfully outside of artificially formal domains, such as crypto arithmetic or theorem proving, have been revolutionary. Revolutionary methods consider examples simultaneously depending upon coincidence among observations rather than programmer-supplied structure to identify rules.

As weak methods, revolutionary procedures cannot deal with representations of the complexity handled by evolutionary procedures. Conversely, evolutionary methods are inapplicable if knowledge of the domain and a formal problem space are not already available.

Problems in Logical Generalization

Procedures for logical generalization identify well-formed formulas (wff's) which are consistent with a set of assertions (data) in the description language. Rule identification is a procedure for determining equivalence classes within which relations are consistent with respect to some criterion.

Selection of appropriate rules from among the generated wff's will in general be equivocal. Problems inherent in the description of data through logical generalization are illustrated using a simple description language, VL1. Figure 1 presents a set of data from which the generalizations shown in Figure 2 can be derived.

	Predicate classes		criterion
	color	flies	
(raven)	black	yes	bird
(cardinal)	red	yes	bird
(penguin)	black	no	bird
(Irish setter)	red	no	dog

Figure 1.

Generalization of these observations leads to two equally preferred rules for identifying birds:

black(obj) -> bird

flies(obj) -> bird

Figure 2.

Equivocality

Consider the case in which all predicate classes have the same number of predicates.

N = number of predicate classes
n = number of predicates/class

Then

n^N = maximum number of discriminable observations

$$\left[\sum_{k=1}^{n} \frac{n!}{(n-k)!k!} \right]^N = \text{number of VL1 wff's}$$

Except for the degenerate case in which there is only one predicate per class (i.e. only one differentiable example and description) the number of descriptions will be strictly greater, typically much greater, than the number of examples. While this equivocality is tempered by accepting only valid (consistent in the criterion) wff's and its exact form will depend on the language used, there will almost always be many more valid descriptions than observations. As a consequence, description by logical generalization revolves around choosing the "best" of the descriptions. Usually staggered criteria such as choosing wff's of greatest coverage and from among these those of greatest parsimony, can be employed. In cases such as the bird example, this choice will be indeterminate.

Errors

Procedures based on logical generalization are inherently intolerant of errors. Valid wff's exclude them by definition. This is not too damaging when inconsistencies occur only on borders between rules or there are many replications of observations which are identical in their noncriterion predicates. Errors along borders may be expected where there is a graduated transition between equivalence classes. Identification of rules separated by undescribed observations accurately reflects this situation. In cases with many replications, imaging, a procedure for assigning a single criterion predicate to unique combinations of predicates, can be employed to filter out error.

For human performance data, however, where occasional errors well within the conditions of the rules being employed by a subject are expected, a problem is posed. Prior to identification the rule is not known so the response is not classified as an error. Subsequent to identification, rules have been gerrymandered to exclude the error so it again eludes detection.

One possible solution is to relax the identification criteria to allow a small number of errors. An experimental version of Michalski's Aq algorithm [5] was implemented to examine this approach. At considerable increase in run-time, the program identified rules of increased coverage (maximum coverage 79 vs. 41) frm PLANT data. Cross validation of these rules, however, revealed performance markedly inferior (errors 7 vs. 29) to that of rules identified using the logical consistency criterion. This decrease in predictive validity is probably attributable to the much larger number of allowable wff's and the attendant increase in equivocality.

A pragmatic approach was ultimately adopted in the analysis of the CDTI data. Generalizations were run using various reduced sets of predicate classes. Identified rules were then selected from among these generalizations based on performance.

A related problem in the identification of rules from human performance data arises in circumstances in which alternate responses may be equally appropriate. Groups of observations for which no rules are found may be indicative of this situation. No solution short of explicitly re-representing alternate responses as additional response categories appears available to data-driven identification methods.

Intersecting Rules

As illustrated in the "bird" example, the set of valid wff's will commonly contain rules sharing observations in common. Such intersections need to be minimized in the selection of a representative rule set so that the data can be described with a relatively few number of rules.

The extent to which some common core of observations is shared by a number of rules serves as an indication of limitations in the language for describing apparent equivalence classes. The structure of the relationship among intersecting rules can be revealed by organizing them in terms of these intersections.

In the analysis of the CDTI data, rules were organized into trees in which 90% of the observations covered by a descendent were covered in common by its parent. If rule identification is pursued with the purpose of explicating observations, this technique provides the additional advantage of graphically portraying the relation among rules. Variations attributable to individual differences or experimental manipulations then become evident. Interpretation is similar to that used with hierarchical clustering procedures.

In the CDTI data, for example, a subgroup of pilots was found to consistently maneuver horizontally toward intruders approaching too closely horizontally. The remainder of the pilots followed the correct strategy of turning toward only those intruders who were both too close horizontally and would pass behind. The rule followed by the errant pilots can be interpreted as a "version" of the correct strategy with reduced dimensionality.

In rule sets generated using reduced sets of predicates the root rule, while providing the greatest coverage, may not be the best performing due to errors introduced through imaging. Better performing descendents with minimal redundant coverage can be selected from the tree by choosing rules from different branches.

Significance Testing

Logical generalization has been introduced as a process of describing data. To stop here, as machine learning programs do, amounts to little more than fitting a "form gauge" to the data. Inductive inference requires both a consistent description and some measure of confidence that the description embodies actual relations rather than idiosyncrasies of the data/identification method. The syntactic rigidity of GLM reduces this problem to estimating sampling error. Even here, when misapplied as in the case of regressions drawing upon large numbers of variables [2], opportunism in sampling from a space of descriptions may result in unwarranted inferences.

For logical induction the hazards are reversed. The number of consistent statements which can be made about a set of observations is much greater than the number of observations. As a consequence expressions of confidence must explicitly consider sampling in a space of descriptions. Christensen and Eilbert [1] refer to this problem as contrivedness, "...the tendency of a search procedure to uncover apparent patterns where none exist."

A logical benchmark to gauge our confidence is the situation in which there is no relation between the conditions of a rule and the associated response. If this were true then any pairing of observed conditions and responses would be equally likely. A distribution corresponding to this benchmark can be obtained by considering all permutations of responses with respect to the conditions. The relative frequency of permutations in which rules of equal or greater "quality" are identified provides a measure of the unlikeliness (significance) of identifying rules of the "quality" obtained under a null hypothesis of no relation.

A nonparametric analogue to the coefficient of determination, tau-b, is representative of the type of statistic often misused in this context, as was shown in Toussaint's review [11]. Significance levels for the associated G-square approximation of chi-square become deceptive when identification methods susceptible to contrivedness have been employed. Figure 3 shows significance levels for tau-b based on a 1000 iteration Monte Carlo procedure which replicated the identification process used in the analysis of the CDTI data. The corresponding significance level based on chi-square is $p < .01$ for all values of tau-b greater than .05.

Monte Carlo Significance Levels for Horizontal-Toward Response

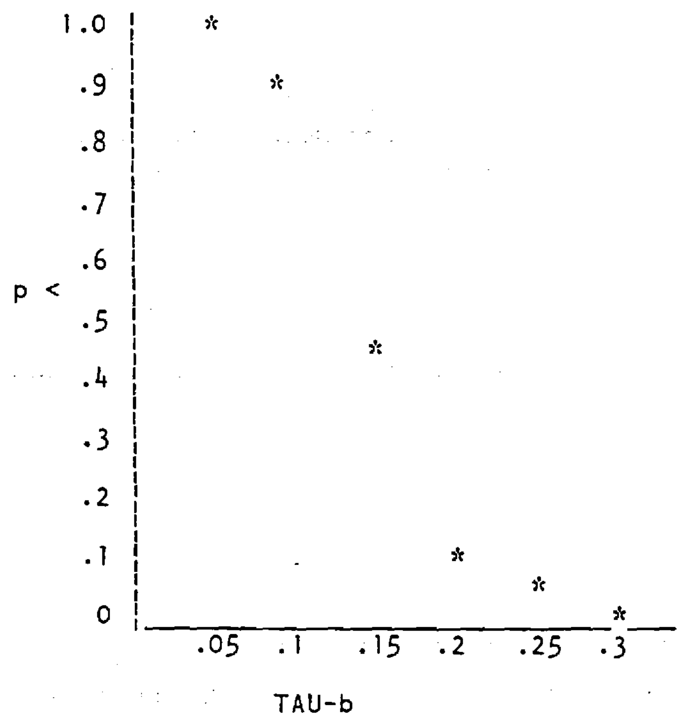


Figure 3.

Knowledge-Constrained Rule Identification

From the problems described in identifying rules, it is apparent that logical generalization fails to generate generalizations of the "quality" that we, as inducers, are accustomed to making. The nature of this difficulty is pointed out by Hume who portrays induction as a psychological rather than a logical process. According to this view, induction is characterized by the repeated pairings of events leading to the *habit* of expecting an event of one sort to be followed by that of another. This formulation identifies rules through coincidence by a recursive process in which there are habits of forming habits. Laws of induction, such as Mill's methods, are subsumed in this manner as "meta habits".

Evolutionary methods mechanize the final step of the recursion process but with less knowledge than the human analyst. A viable identification methodology for rule-based models should incorporate the problem reduction available to evolutionary methods, while preserving the objectivity and possibility of novelty accorded the revolutionary approach.

A compromise is afforded by taking the Bayesian view of rule identification. Knowledge/assumptions could be made explicit, and rule identification could be conditioned on these assumptions. Constraining the generation of rules would result in reduced equivocality with accompanying increase in power for conditional significance tests. Experimenters could set constraints at levels appropriate to research goals. Reporting these constraints as Bayesians do their priors would make explicit the contribution of data to the results. As rule-based modeling comes of age, conventions such as these will be required, if a coherent body of knowledge is to be developed.

Acknowledgment

This research is supported by the National Aeronautics and Space Administration under Ames Grant No. NAG 2-123.

References

[1] R.F Eilbert, and R.A. Christensen, "Contrivedness: the boundary between pattern recognition and numerology", Pattern Recognition, pp. 253-261, 1982.

[2] A.B. Forsythe, J. Laszlo, K., Jennrich, and P.R.A. May, "A Stopping Rule for Variable Selection in Multiple Regression", Journal of the American Statistical Association, vol. 68, pp. 75-77, 1983.

[3] R.W Hunt, and W.B Rouse, "A fuzzy rule-based model of human problem solving", IEEE Transactions on Systems, Man, and Cybernetics, vol 14, pp. 112-119, 1984.

[4] R.S. Michalski, "Knowledge repair mechanisms: evolution vs. revolution", in Proceedings of the Third International Machine Learning Workshop, Skytop, Pennsylvania, June 24-26, 1985.

[5] R.S. Michalski, "On the quasi-minimal solution of the general covering problem", in Proceedings of the Fifth International Symposium on Information Processing, A3, Bled, Yugoslavia, 1969.

[6] R.S Michalski, "Pattern recognition as rule guided inference", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 4, pp.349-361, 1982.

[7] W.M. Morris "Human problem solving in process control" (Tech. Rep. 83-2). Atlanta, Georgia: Georgia Institute of Technology, Center for Man-Machine Systems Research, August, 1983.

[8] A. Newell, and H. A. Simon, Human Problem Solving. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

[9] E. Palmer, "Conflict resolution maneuvers during near miss encounters with cockpit traffic displays", in Proceedings of the Human Factors Society-27th Annual Meeting, Santa Monica, Ca.: Human Factors Society, pp. 757-761, 1983.

[10] H.B. Simon and G. Lea, "Problem solving and rule induction", in W. Gregg (Ed.), Knowledge and Cognition. Hillsdale, NJ: Lawrence Erlbaum Associates, 1974, pp. 105-128.

[11] G. Toussaint, "Bibliography on estimation of misclassification", IEEE Transaction on Information Theory, vol. 20, pp. 472-479, 1974.

[12] R.B Wesson, "Planning in the world of the air traffic controller", Proceedings of the 5th International Joint Conference on Artificial Intelligence, M.I.T., Cambridge, Mass., 1977, pp. 473-479.

REPRESENTATION for CLOSED FORM SIGNIFICANCE TESTING in VL1

C. Michael Lewis

Center for Man-Machine Systems Research

School of Industrial and Systems Engineering

Georgia Institute of Technology

(A Working Paper)

Programs for machine learning and systems for logical induction such as that of Carnap share certain characteristics:

> A language, L, consisting of a vocabulary of predicates and connectives is defined for describing a set of observations. Induction is achieved by identifying generalized statements (wff's) describing the observations. In this paper these statements will be referred to as "rules".

As relations among predicates are restricted to legal connectives (usually: &, V, ~), inductive inference, based on the coincidence of predicates rather than the correlation of values, results. This distinction is basic to the problem of inference since relations are identified between logical phrases and their consequents rather than between individual "variables".

In machine learning and other applications of inductive logic this usually reduces to a discriminant procedure. Predicates are divided into a group describing the observations and a class of mutually exclusive discriminant predicates one of which is designated as the positive case, '+', while the others are treated as negative cases. When used to describe behavioral 'rules', phrases describing the stimuli which are accompanied by a particular response are discriminated from all others.

While Carnap's system, based on degrees of confirmation, allows rules to describe negative instances, machine learning programs typically inspect only valid rules to restrict search.

If this logical generalization is to be used to form inferences from actual observations the relationship between the data and its population must be considered. The primary threat to validity in the identification of rules through combinatorial coincidence lies in the ability of such procedures to identify apparent consistencies (in the sample) where none exist (in the population). This occurs because the generalization procedure can examine the data in so many ways that it is possible to discover a relationship that is only due to chance. To guage our confidence in a particular rule, it is necessary to find some benchmark with which it can be referenced. A logical choice is the situation in which there is no relation between observation descriptions and their classification as '+' or '-,' then any pairing of observation descriptions and discriminant classifications would be equally likely (principle of indifference). If there are N observations and k of these observations are classified as '+' then there are $N!/(N-k)!k!$ distinct mappings from the observation descriptions to the responses. These mappings will be referred to as instantiations.

The rub here is that the possible instantiations need to be expressed in terms of the rules which would be identified rather than the mappings of observation descriptions to responses. themselves.

The GENERAL PROBLEM

Constraints:

1- The syntax of language, L, which defines the ways in which predicates & connectives can be combined to form rules (wff's)

2- The set of observation descriptions which determines the discriminations which can be made

3- The number of observations classified '+'

The Problem:

Devise a method for determining the relative frequency over instantiations with which a rule of equal or greater generality would be identified. Generality, here, is defined as the number of observations described by a rule.

My Problem:

I am using a covering algorithm (Aq, Michalski 1973) to identify pilot strategy (system of rules) in maneuvering to avoid intruders. The Aq algorithm identifies rules in the VL1 (Michalski's terminology) language, a simplified propositional logic. VL1 allows only one predicate per class of predicates to be true in a particular instantiation. VL1 syntax allows only one place predicates. example: black(bird). Disjunction is only allowed within classes of predicates. example: black(bird) V blue(bird). Conjunction is only allowed between classes of predicates. example: black(bird) & is-raven(bird).

For notational convenience Michalski defines a new entity, a selector, to describe classes of predicates. A selector names the predicate class and lists the predicates involved in a disjunction. example: [color-bird=black V blue]. Redefined in terms of selectors, VL1 allows only conjunction of selectors. Since generalization of the observation descriptions is performed to discriminate the '+' classification, this relationship is represented as An example of a wff in VL1 is:

[x1=a V b] & [x2=d V e]-> F

This rule says that if the predicate evaluating true for class x1 is a or b and class x2 is d or e then F(observation) is true.

An advantage of VL1 for devising a significance test is that this syntax is highly restrictive making the number of possible rules to be considered relatively small. For example a rule to cover the observation descriptions:

[x1=1] & [x2=1] & [x3=1]

[x1=1] & [x2=2] & [x3=1]

[x1=2] & [x2=1] & [x3=1]

MUST ALSO INCLUDE

[x1=2] & [x2=2] & [x3=1]

To produce a generalized conjunction of selectors

[x1=1 V 2] & [x2=1 V 2] & [x3=1]

I will refer to this syntactic property as a rectangularity constraint.

Despite the simplicity of the VL1 language and its restrictive syntax there are a very large number of rules which would be identified across the instantiations.

A Tentative Representation:

1- A Monte Carlo approach of repeatedly running the identification program on randomly selected instantiations is inelegant. It also unnecessarily expends resources on instantiations in which rules, less general than those being tested are identified.

2- A solution may be to ennumerate instantiations for which rules as/or more general would be identified had the identification program been run. Since only rules of substantial generality should be objects of testing, reduction in generation should be achieved. A threshold could be set so that the program terminates after ennumerating > N instantiations corresponding to a predetermined significance level, introducing further economy.

Ennumerating Instantiations

The no relation benchmark corresponds to the distribution of the most general rule across instantiations. Since for any particular rule to be identified all observations must belong to the K observations in the '+' class, let k= the number of observations covered by a particular rule.

Eq. 1

(N-k)!/(N-K)!(K-k)! is then the number of instantiations in which that rule is valid.

Obtaining the set of possible rules of generality >= that being tested must be considered. The notion of discriminant equivalences will be introduced to accomplish this.

Definition: Discrimination level- a particular predicate class or conjunction of predicate classes. For example [x1=a,b....] would be a selector for a predicate class, [x1=a.b,...] & [x2=c,d....] would be a selector for a conjunction of predicate classes.

Definition: Discriminant equivalence

Let M be a discrimination level. A discriminant equivalence, dq(M in level, M, is a set of observation descriptions having identical predicate or conjunction of predicates for the predicate or predicate classes of M.

Due to the syntax of VL1 (conjunction of selectors), rules can only be formed at a single level if the degenerate case of a selector specifying all members of a predicate class is excluded. As a consequence any rule at level M must either include or exclude all members of a dq(M.i) at level M and the dq(M.i)'s at each level represent a complete partitioning of the observation descriptions. As a consequence:

1- discriminations made by any rule at level, M. may be represented as a conjunction of dq(M.i)'s at level M.
2- All discriminations among observations in VL1 are covered if all levels are represented in this way

Since knowing the number of observation descriptions described by a rule allows the ennumeration of instantiations for which it is valid. (Eq.1). this provides a basis for forming and counting rules in accordance to generality across instantiations.

We, however, are interested in counting instantiations for which rules of >= generality than that being tested would be identified. Since it is possible for a rule to be valid yet not be the most general identifiable rule for a particular instantiation, the ennumeration must be adjusted. A rule of greater generality will be said to dominate a rule of lesser generality for instantiations for which both are valid. For example: any rule at the same level which includes an additional dq(M.i) will dominate. Example: [x1=1,2] & [x2=2] dominates [x1=2] & [x2=2], the dq(M.i)'s in this instance are observation descriptions for which x1=1 & x2=2 and observation descriptions for which x1=2 & x2=2.

Between Level Dominance

Definition: Dominance set, Dq(L,M,i)

> A dominance set, Dq(L,M.i), at level L of dq(M.i) at level M is defined as the minimal set of dq(M.i)'s at level M that contains all of the observation descriptions contained by dq(M,i) and is of cardinality < K, the number of observations in the '+' class

Let N= number of observation descriptions

K= number of observations in class '+'

$|dq(M,i)|=k$

$|Dq(L,M,i)|=k^*$

THEN

$(N-k)!/(N-K)!(K-k)! - (N-k^*)!/(N-K)!(N-k^*)!$

is the number of instantiations in which dq(M.i) is not dominated by any rule from level M.

Minimal dominance rules at level, L, for rules at level. M, (conjunctions of dq(M.i)'s) are then simply the union of the corresponding Dq(L,M,i)'s at level L (with additional dq(L,i)'s at level L included as required by rectangularity constraints). Undominated instantiations can be ennumerated in the same way as before.

To find the undominated instantiations of a rule across all levels, however, requires consideration of co-dominance and multiple dominances as well.

example:

A rule at level L is dominated by its corresponding minimal dominance rule at level M. This rule may in term be dominated by its own minimal dominance rule at level N... This problem can be dealt with without recursion by considering all unions for a set of minimal dominance rules (discarding those > K) for the initial rule at level, L, and applying the inclusion/exclusion principle.

To use this representation in deriving a significance test for VL1 rules will require:

1- Some way to ennumerate instantiations for >= rules directly for the dq(M.i)'s and Dq(L,M.i)'s without recourse to actually forming the rules (tagged generating functions?)

failing this

2- Some computationally cheap way to find the undominated instantiations of a rule without having to consider all unions of its minimal dominance rules

A Discrete Control Model

of

PLANT

Christine M. Mitchell

Supported by NASA-Ames Grant #NAG-2-123

December 1984

Introduction

This task entailed the development of a model of the PLANT system using the discrete control modeling techniques developed by Miller (1985). Discrete control models attempt to represent in a mathematical form how a human operator might decompose a complex system into simpler parts and how s/he coordinates control actions and system configuration so that acceptable overall system performance is achieved. Basic questions include knowledge representation, information flow, and decision making in complex systems. The structure of the model is a general hierarchical/heterarchical scheme which structurally accounts for coordination and dynamic focus of attention. Mathematically, the discrete control model is defined in terms of a network of finite state systems.

The discrete control model can be thought of as a possible representation of an operator's internal model of the system plus a control structure which specifies how the model is used to solve the decision problems which make up the control functions. Specifically, the discrete control model accounts for how specific control actions are selected from information about the controlled system, the environment, and the context of the situation. The objective is to provide a plausible and empirically testable accounting and, if possible, explanation of control behavior.

Theoretical details and practical mechanics of discrete control modeling are detailed in Miller (1985) and Mitchell (1980). The model described below assumes most of this material as background.

Model Preliminaries

The first step in constructing a discrete control model is to specify the lowest level of description, the output of the model and the bottom nodes in the finite state network. Several discrete control models have based their structure on configurations of system switches (Miller, 1979; Mitchell, 1980). A model like this for PLANT, for example, would utilize the configuration of valves and the flow of resources (i.e., PI and PO) through the system. The initial model development for PLANT in fact began at this point. As modeling progressed, however, it became clear that the more interesting output of a PLANT discrete control model was the operator commands which were employed to configure and optimize PLANT. The commands available to the operator are summarized in table 1.

Using operator commands as the lowest level finite state nodes of the discrete control network, the model attempts to explain an operator's choice of commands based on system state and current operator function or procedure. The model is normative in that it is constructed using both the rules of the system as specified in PLANT documentation and the procedures provided to PLANT operators (Morris, 1983). Such a model could be used to "explain" operator behavior, that is, to justify and contextualize a sequence of operator actions based on goals and objectives. In addition, a normative discrete control model may be useful in designing an adaptive user interface that is responsive to user needs. Additional details on applications of the PLANT discrete control model follow the presentation of the model itself.

ovI,J	Open the valve between tanks I and J
cvI,J	Close the valve between tanks I and J
ocK	Open one valve per tank in column K
ccK	Close one valve per tank in column K
otI	Open all valves from tank I
ctI	Close all valves from tank I
piN	Set input per input tank to N units
poN	Set output per output tank to N units
skN	Skip N iterations; the system will be updated N times before the display is updated
flI,J	Check the flow from tank I to tank J
afI	Check all flows from tank I
rvI,J	Repair the valve between tanks I and J
rpI	Repair the pump associated with tank I
rtI	Repair the rupture of tank I
rs	Repair the PLANT safety system
st	System trip; close all valves and stop all input and output

Table 1. PLANT Operator Commands

The PLANT Control Network

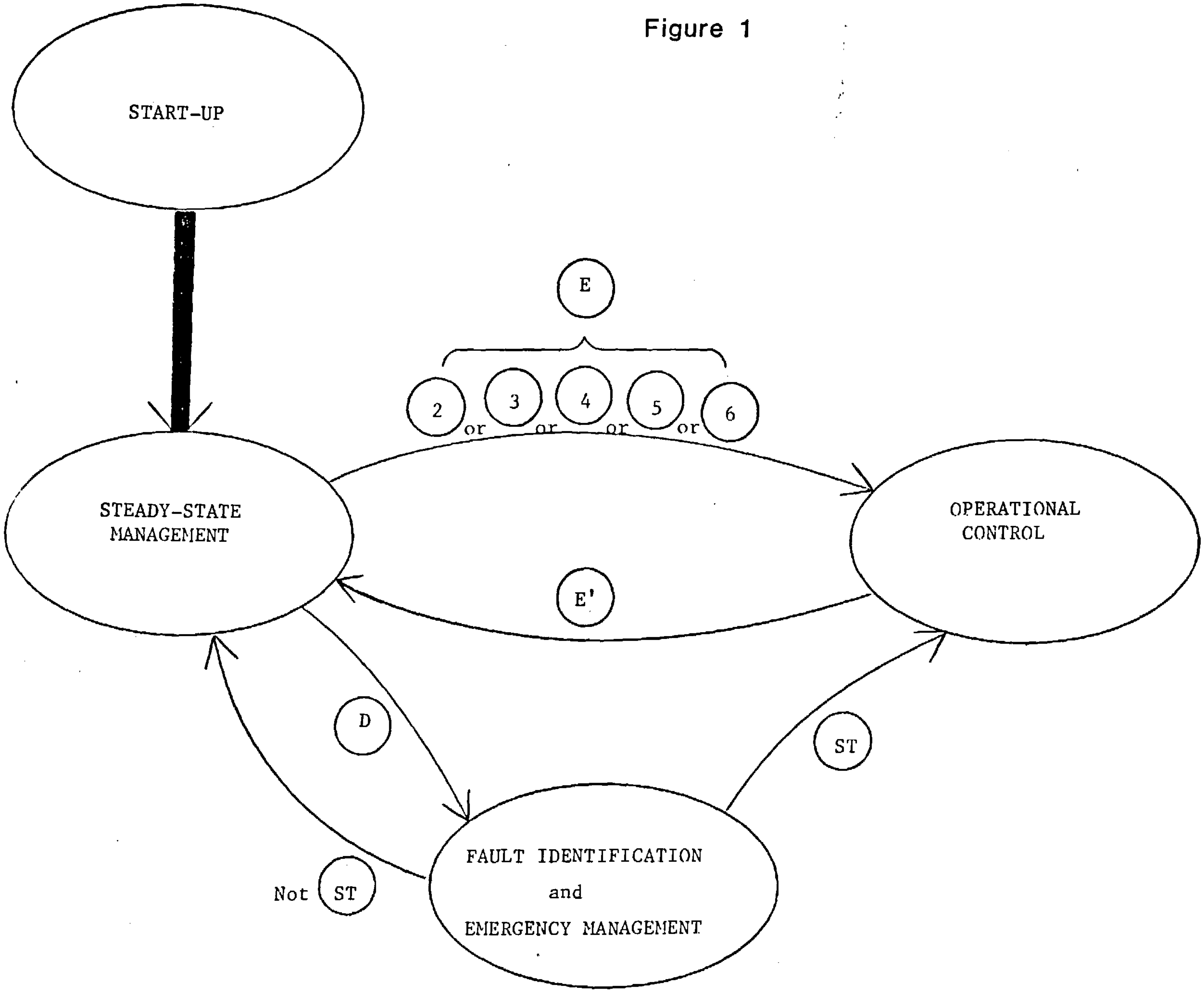
At the highest level, there are four major operator control functions for PLANT. As depicted in figure 1, the first control function that an operator engages in is system start-up. Once this set of activities is completed the operator unconditionally transitions to the function of steady-state management. For PLANT, steady-state management is a monitoring and fault detection state. In this state, the operator is essentially a supervisory controller, watching a fairly autonomous system operate within the boundaries specified by the system configuration.

From steady-state management, an operator can, under certain conditions, transition to a more active control function. If a fault is suspected or if the system is approaching an out-of-control condition, the operator engages is fault identification or emergency management. On the other hand, while engaged in steady-state management the operator may notice the evolution of one or more symptoms which suggest the need for proceduralized manual control.

Figure 1 and the associated footnotes depict these transitions. Several conventions are used in this figure. The heavy black arc between the startup node and the steady-state management node is used to denote an unconditional transition. This convention will be used both in figure 1 and in subsequent figures. Conditional arcs are those that denote state transitions which occur only when enabling conditions are met. For example, steady-state management shifts to operational control when one of the conditions calling for a procedure are met.

PLANT – Control Overview

Figure 1



Control Overview Notes & Symbols

1. The heavy black arrow from start-up to steady-state management denotes an automatic or unconditional transition, i.e., once the start-up control function terminates there is an automatic transition to the steady-state management function. Moreover, as the figure suggests, the start-up function is performed exactly once for the control session.

2. Control shifts from start-up to steady-state management and then to fault identification and emergency management when a fault or operational problem is suspected.

   D : fault suspected due to
   - random tank check
   - drop in resources
   - insufficient number of system trips
   - unmanageable number of system trips

   E or i : symptoms are present which call for the use of procedure i, i = 2,3,4,5,6

3. Control shifts from operational problem solving to steady-state management when system is reconfigured at a minimum stabilization E'. E' will be further specified in the operational control function.

4. ST indicates an operator-initiated system trip.

The conditional arcs at this level of the network are straightforward. Condition E is an enabling condition which is true when the system state requires the use of a prespecified procedure. Condition E' is a minimum level of system stabilization; this condition is set to true at the completion of each procedure in operational control. Condition D is true when a system fault is suspected during routine management or if the number of system trips is so excessive as to make the operator feel that PLANT is in an out-of-control condition. Finally, condition ST indicates that the operator initiated a system trip.

The structure of discrete control models is both hierarchic and heterarchic. The portion of the model depicted in figure 1 is at the highest level of the hierarchy and depicts the somewhat heterarchic activities that take place at this level. For the PLANT model, the next level of detail explores particular activities or subfunctions within each of the major control activities previously described. Figures 2-9 constitute the remainder of the model.

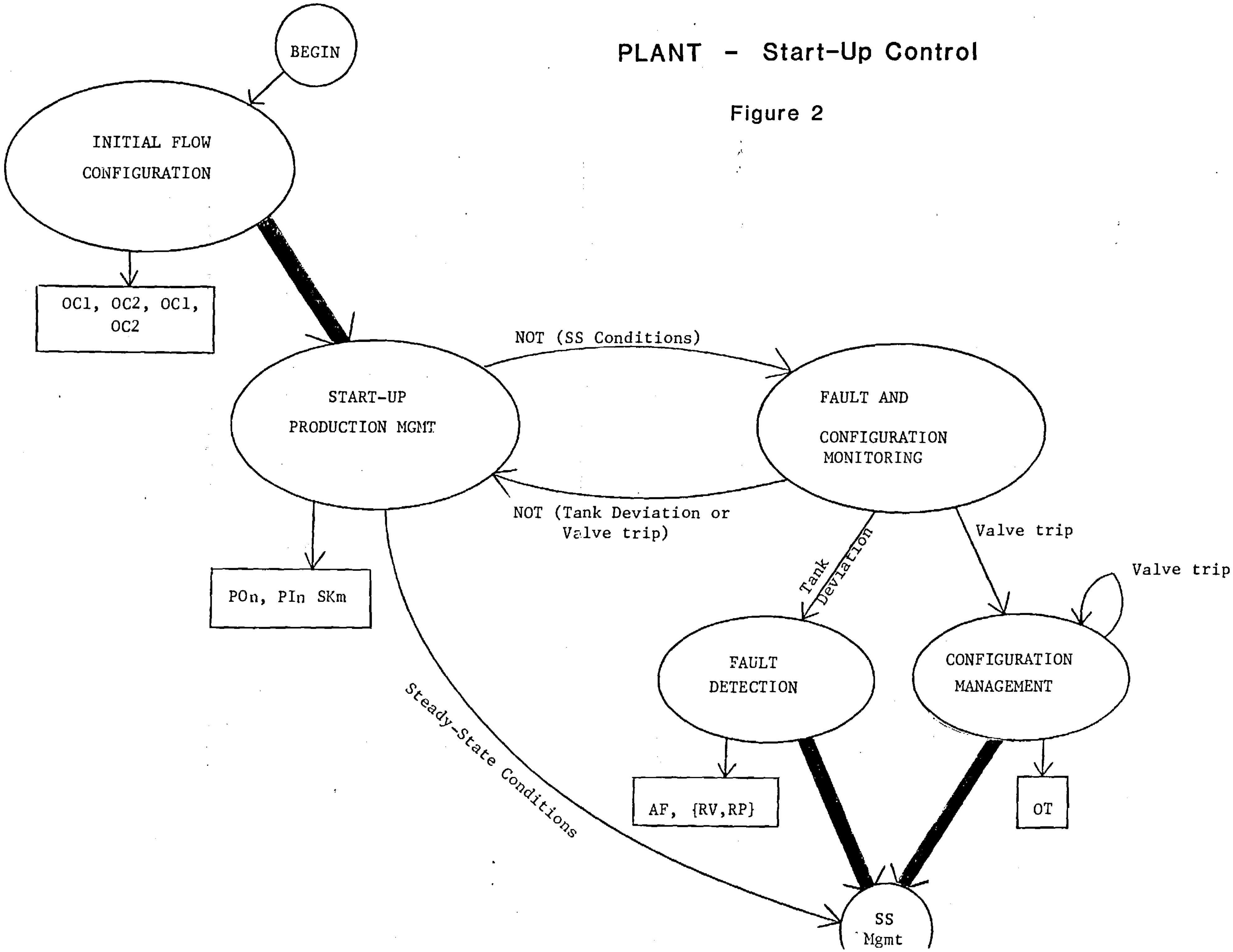
Start-Up Control

The start-up control function (figure 2) consists of three major subfunctions. Initially, the operator opens the valves among the tanks with a sequence of commands that completes the initial flow configuration.

The conventions used in figures 2-11 include the use of elliptical nodes for operator functions or actions and rectangular boxes for commands. The distinction being made is one of activity versus intention. The commands are activities undertaken to accomplish an objective of a function.

PLANT - Start-Up Control

Figure 2



Start-Up Control Notes

1. Steady-state conditions: pi = po = 230

2. Tank deviation: inequality of tank heights within columns

3. Start-up production management: n = 50,100,150,200,210,220,230;
   m = 5

Once the initial flow configuration is completed the operator unconditionally transfers to a start-up production management function, the function which increases input and output to a steady-state condition. Each start-up production management task increases output (PO), input (PI), and may skip (SK) one or more PLANT iterations. Once a round is accomplished the operator checks to see if steady-state conditions have been met, i.e., PI = PO = 230. If so, the start-up function is concluded and the operator unconditionally transitions from start-up control to steady-state management. If start-up conditions have not been met, the operator transitions to a fault and configuration monitoring subfunction, in which tank heights are scanned to ensure levels are the same within columns and no valve has tripped. If a tank deviation or valve trip has occurred the operator performs the appropriate remedial action, i.e., opens the tripped valves or checks flows to identify and fix the failed valve or pump. Once a fault or valve trip has been identified the start-up control function concludes and the operator transitions to the steady-state management function. If after checking tank levels and valves, no problems are detected the operator again commences on another start-up production management subfunction.

To summarize, the start-up control function terminates in one of two ways. Typically, start-up control is completed when the operator has configured PLANT into a minimally stable and optimal mode. If, however, a fault is detected before start-up is complete, diagnostic and compensation procedures are performed and the operator terminates start-up control and engages in steady-state management.

Steady-State Management

Following the completion of start-up control, steady-state management is the next high level control function undertaken (figure 3). This control function has three major components: monitoring and fault detection, configuration management, and production optimizaton.

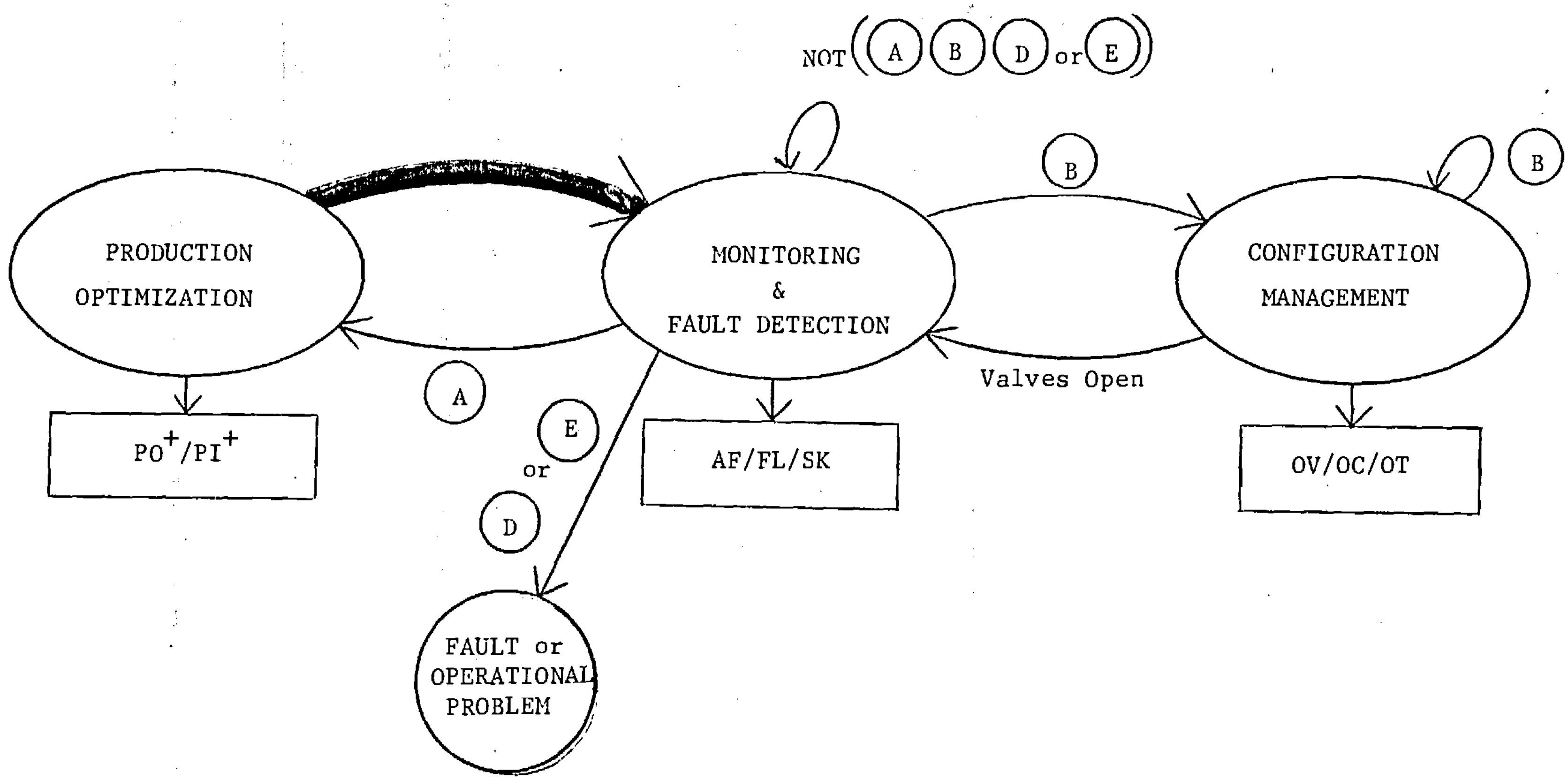
Monitoring and fault detection is the central subfunction. An operator may perform this function with visual scans of the PLANT displays together with iteration skips (SK) or by actively testing tank flows (AF,FL). The operator terminates this function to undertake another steady-state management subfunction if one or more valves trip, if a repair crew completion message arrives, or if configuration is acceptable and production optimization is required. Configuration management is the subfunction which keeps all valves open under normal conditions and reopens valves after repair completion. Product optimization entails one or more commands to balance and/or increase input and output. This subfunction is pursued when the PLANT is stable and production is less than the specified goal, i.e., PO = PI = 230. Operator control remains in the steady-state management function, alternating among its subfunctions, until some problem arises. Operator control leaves the steady-state management function when a problem requiring either proceduralized control or fault detection occurs.

Operational Control

Operational control is an operator function which is, as compared to steady-state management, proceduralized and low level; in this function, the operator exercises a great deal of direct manipulation over the system (figure 4). Depending on the symptoms, the operator engages in

PLANT - Steady-State Management

Figure 3



A Input/Output Tuning Required (pi ≠ po or pi < 230 or po < 230) and low frequency of valve trips and tanks are within acceptable range*)

B Isolated valve trip or repair crew completion

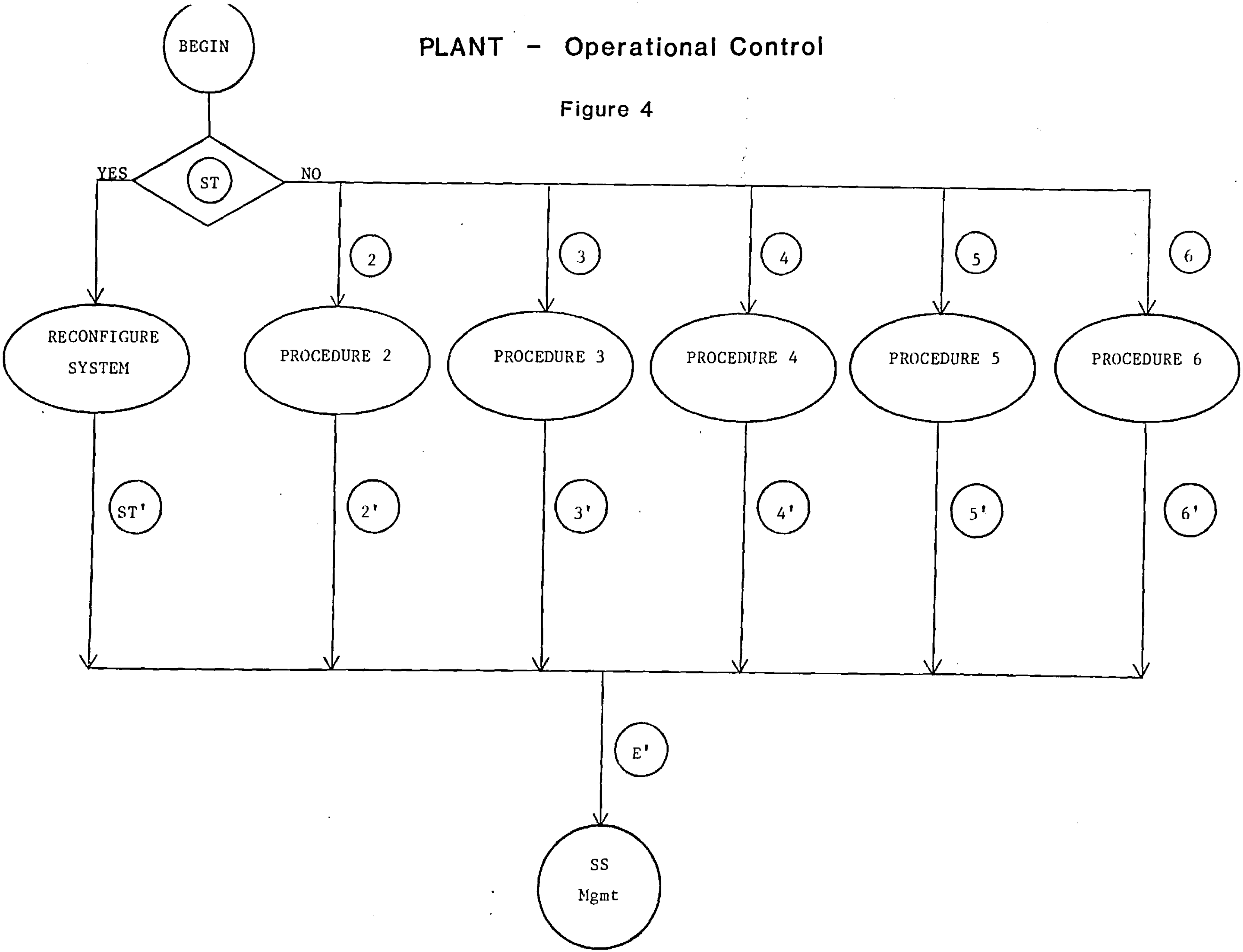
D Fault suspected (see control overview)

E Symptoms of need for proceduralized operation (see control overview)

* largest difference < 30 units, all levels are > 10 and < 90 units

PLANT - Operational Control

Figure 4



Operational Control Notes

1. (ST) indicates an operator-initiated system trip

2. (i) symptoms present which require use of procedure i, i = 2,...,6.

3. (ST') or (i') conditions have been met which terminate the procedure, control then shifts back to steady state management.

4. (E') system stabilized E' = T if (ST') or (i') for i = 2,3,4,5,6

one of the prespecified control procedures (see Morris, 1983) or in reconfiguration after a system trip. When an operational control procedure ends, operator control returns to steady-state management after a minimum system stabilization point is reached.

The simplest subfunction in operational control is PLANT reconfiguration after a system trip (figure 5). This simply consists of reopening all valves. Once all valves are opened, operator function transitions back to steady-state management. Input and output production increases are handled by the product optimization subfunction of steady-state management and are accompanied by fault detection and monitoring.

The remaining procedures in the operational control function are similar to those used by Morris (1983) to train PLANT controllers. The symptoms that indicate a need for these procedures are summarized in Appendix A. The PLANT discrete control model has structured the steps in the procedures and, as a result, they are not quite isomorphic to those used in operator training. One major difference betwen Morris' procedures and the model's procedures are the termination points of the procedures. The model's procedures terminate as soon as a minimal point of system stabilization is reached; system optimization, input and output increase for example, are completed as part of the steady-state management function. The result of these changes in terms of operator control activities, however, should be equivalent. The model's version of these procedures is summarized below.

Procedure 2 addresses the problem arising when PLANT's input column tank levels are higher than those of the other columns (figure 6). This procedure consists of three subfunctions: production limitation wherein, depending on system symptoms, input and output are reduced; valve

PLANT - Operational Control

Reconfigure System After Operator-Initiated System Trip

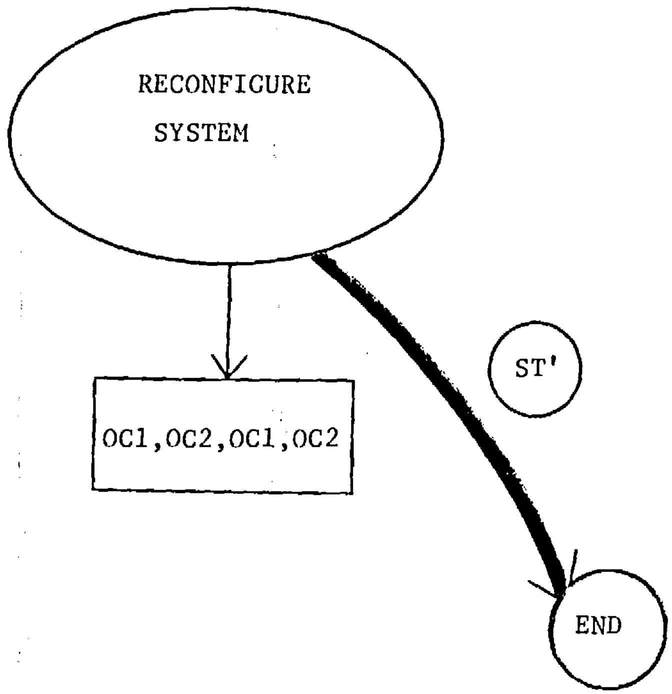
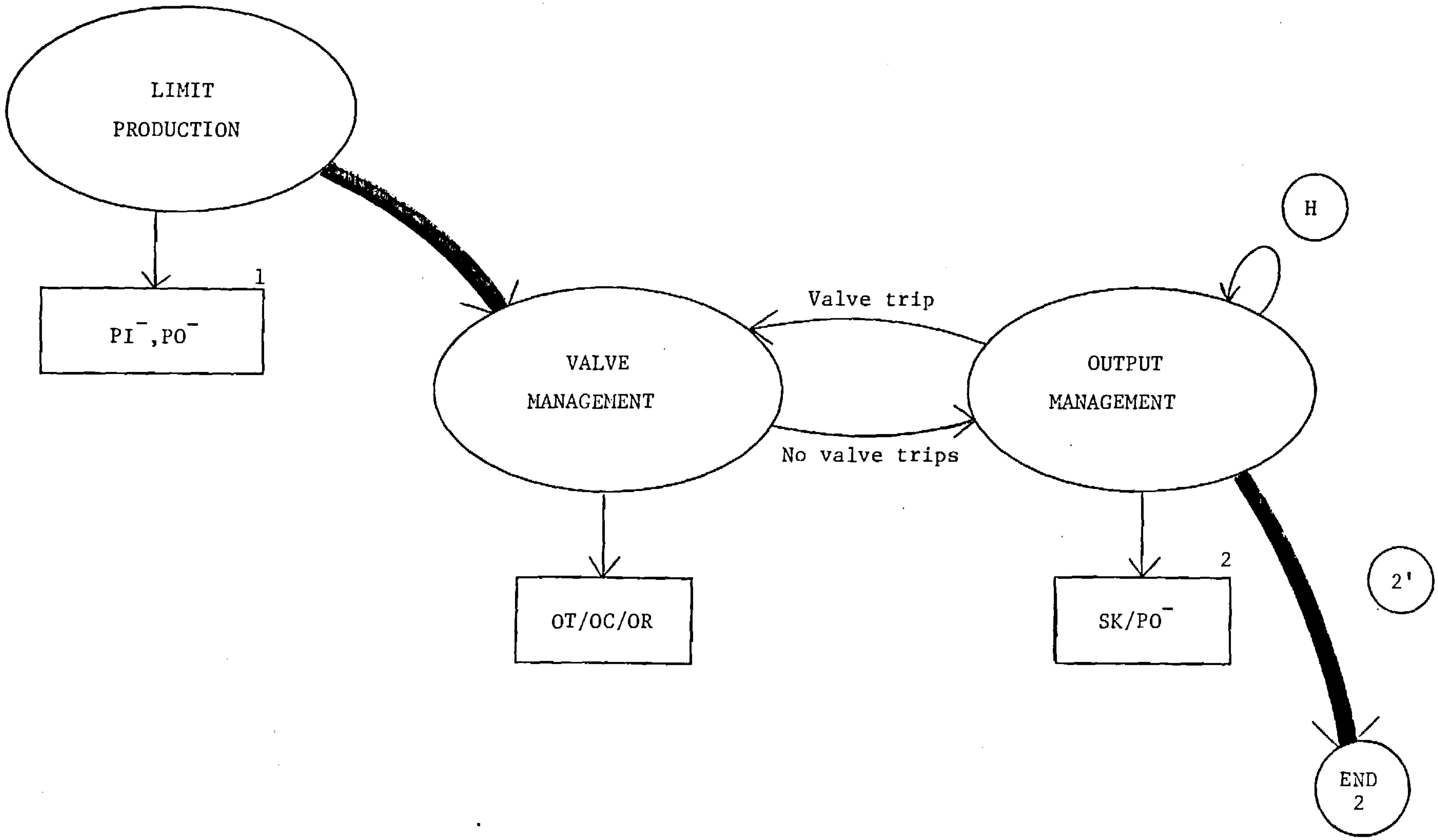


Figure 5

PLANT – Operational Control

Procedure 2 BC Problem for Input Column

Figure 6



PLANT-Operational Control Procedure 2 Notes

1. Rule IF any tank in Col. 1 > 90
   THEN PI: = 0 & PO: = (0 - 25)
   ELSE PI: = 50 & PO: = (50 - 75)

2. Rule IF tanks in Col. 2 and 3 continue
   to drop and iteration = 10
   THEN PO: = Ø

H. PO > Ø AND less than ten iterations

2'. PO: = Ø OR iterations > 10

management which keeps tripped valves open; and output management which monitors output and adjusts production to keep columns 2 and 3 fairly balanced. The procedure concludes when columns 2 and 3 stabilize or if production output is set to zero.

Procedure 3, a condition in which column 3 tank levels are too low, also begins with a production curtailment function and unconditionally transitions to a valve management state (figure 7). From there, the operator monitors valve trips, input, and output until columns all stabilize (i.e., columns 1 and 2 do not continue to increase and column one does not continue to decrease) or until both input and output are set to zero.

Procedure 4 addresses the problems of high tank levels in column 1 and low levels in column 3, problems which combine those of procedures 2 and 3 (figure 8). The steps required in procedure 4 are very simple; first input is reduced, then output is reduced.

Procedures 5 and 6 (figures 9 and 10) address imbalances within columns rather than the imbalance between columns addressed in procedures 2 through 4. Since within column imbalances are often due to system faults, an initial concern is to examine the system for component failures. Procedure 5 first limits input, then examines the high tank for possible valve failure. Following failure detection, the operator also limits output. Finally, the operator engages in a set of monitoring and tuning activities to compensate for the imbalance and/or component failures. Tripped valves are opened. If column 1 shows an excessive imbalance among tanks, valves are temporarily closed. Similarly, if tank levels in columns 2 and 3 continue to decrease over time, output is

PLANT - Operational Control

Procedure 3: BC Problem for Output Column

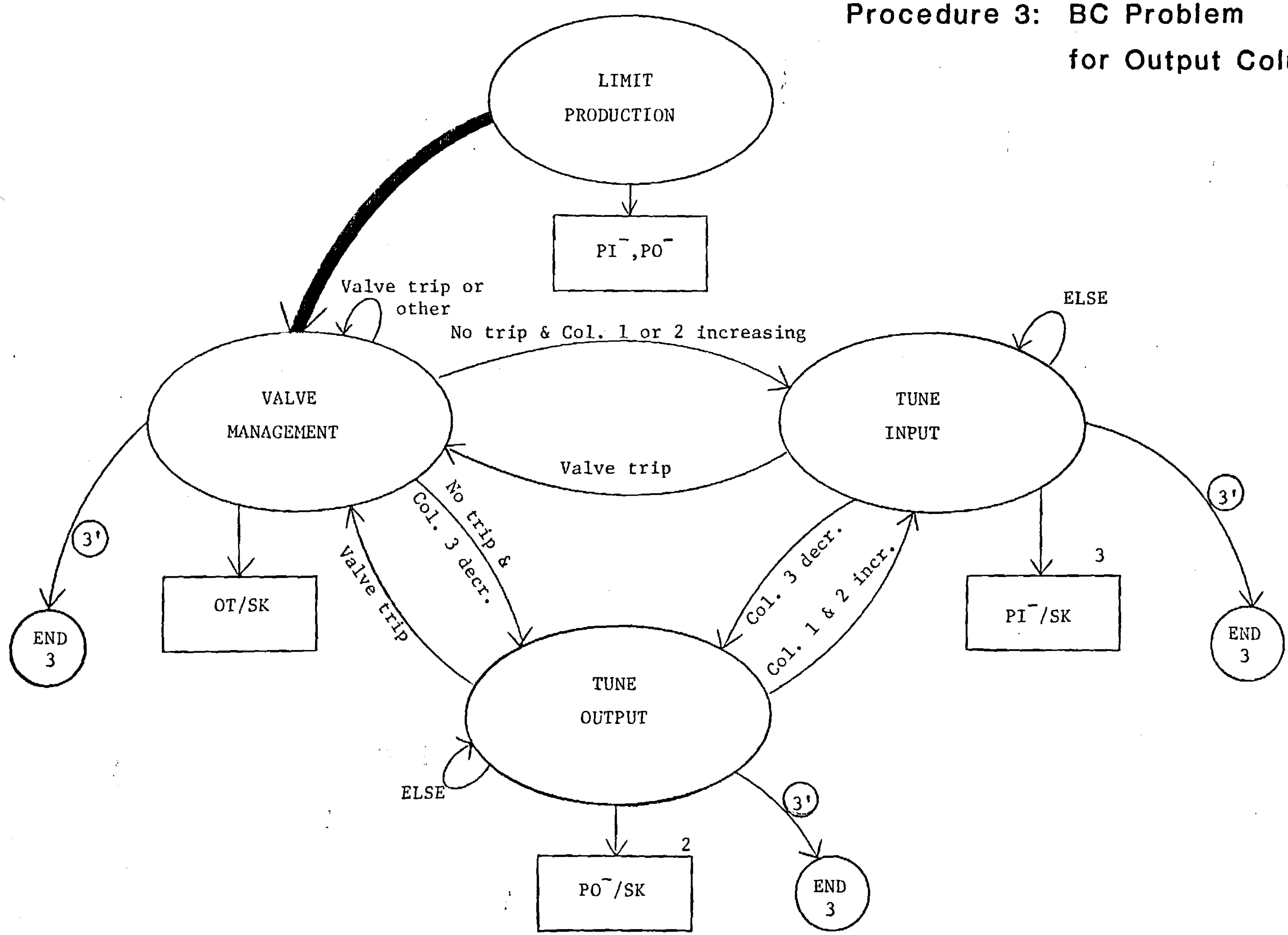


Figure 7

PLANT-Operational Control Procedure 3 Notes

1. PI: = (50 - 100); PO: = 50

2. IF Col. 3 < 5 then PO: = Ø
   ELSE IF Col. 3 < 10 then PO: = 25

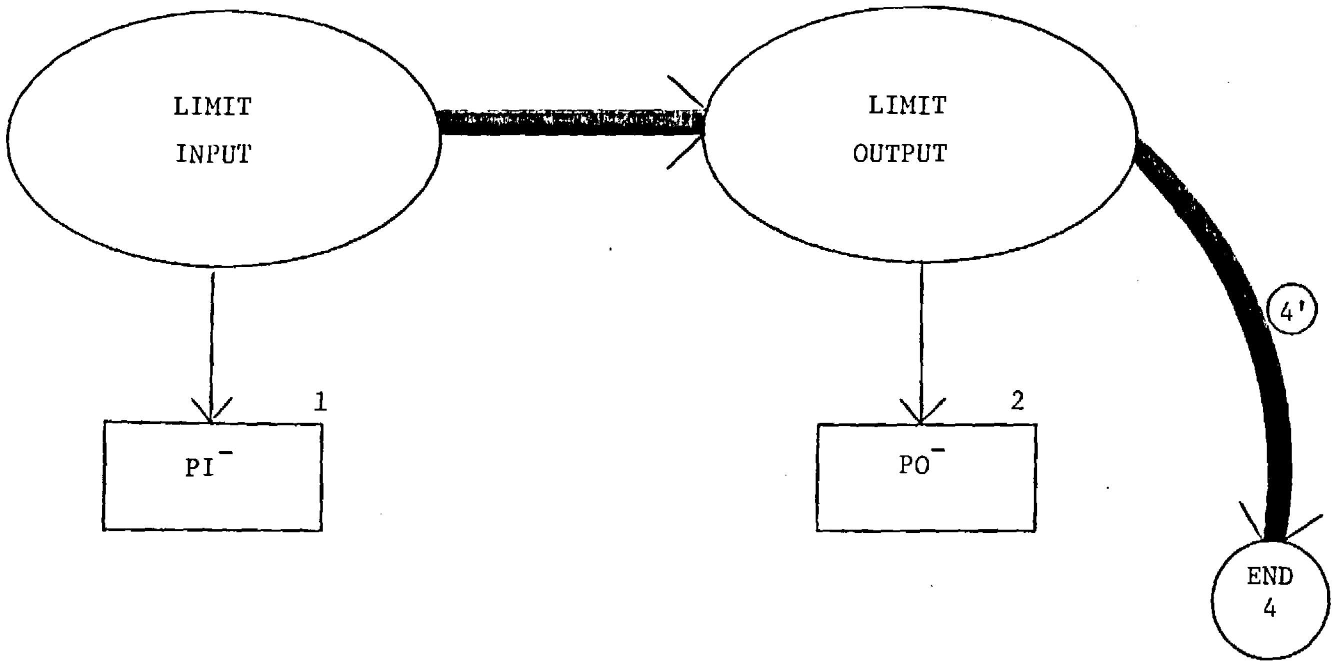
3. IF Col. 1 or 2 increases over 5 iterations, PI: = (25 - 50):
   IF frequent valve trips, PI: = 0

3'. IF PI = PO = Ø or Col. 1 and 2 and 3 are stabilized

PLANT - Operational Control

Procedure 4: BC Problem for Input and Output

Figure 8



1 Rule: IF col. 1 > 90 then PI: = Ø
ELSE PI: = 50

2 Rule: IF col. 3 < 5 then PO: = Ø
ELSE PO: = 50

Figure 9

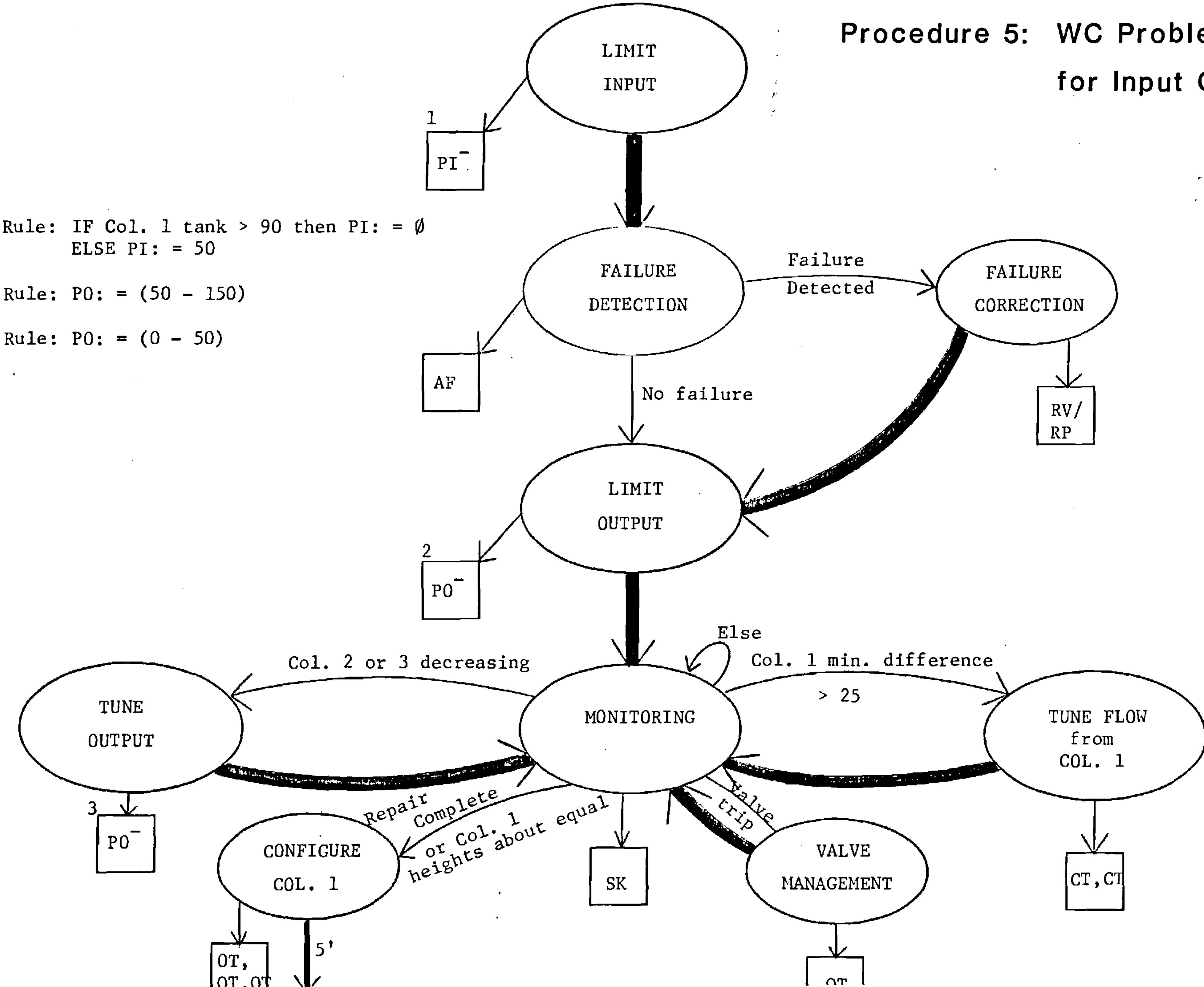
PLANT – Operational Control

Procedure 5: WC Problem for Input Column

1 Rule: IF Col. 1 tank > 90 then PI: = Ø
ELSE PI: = 50

2 Rule: PO: = (50 – 150)

3 Rule: PO: = (0 – 50)

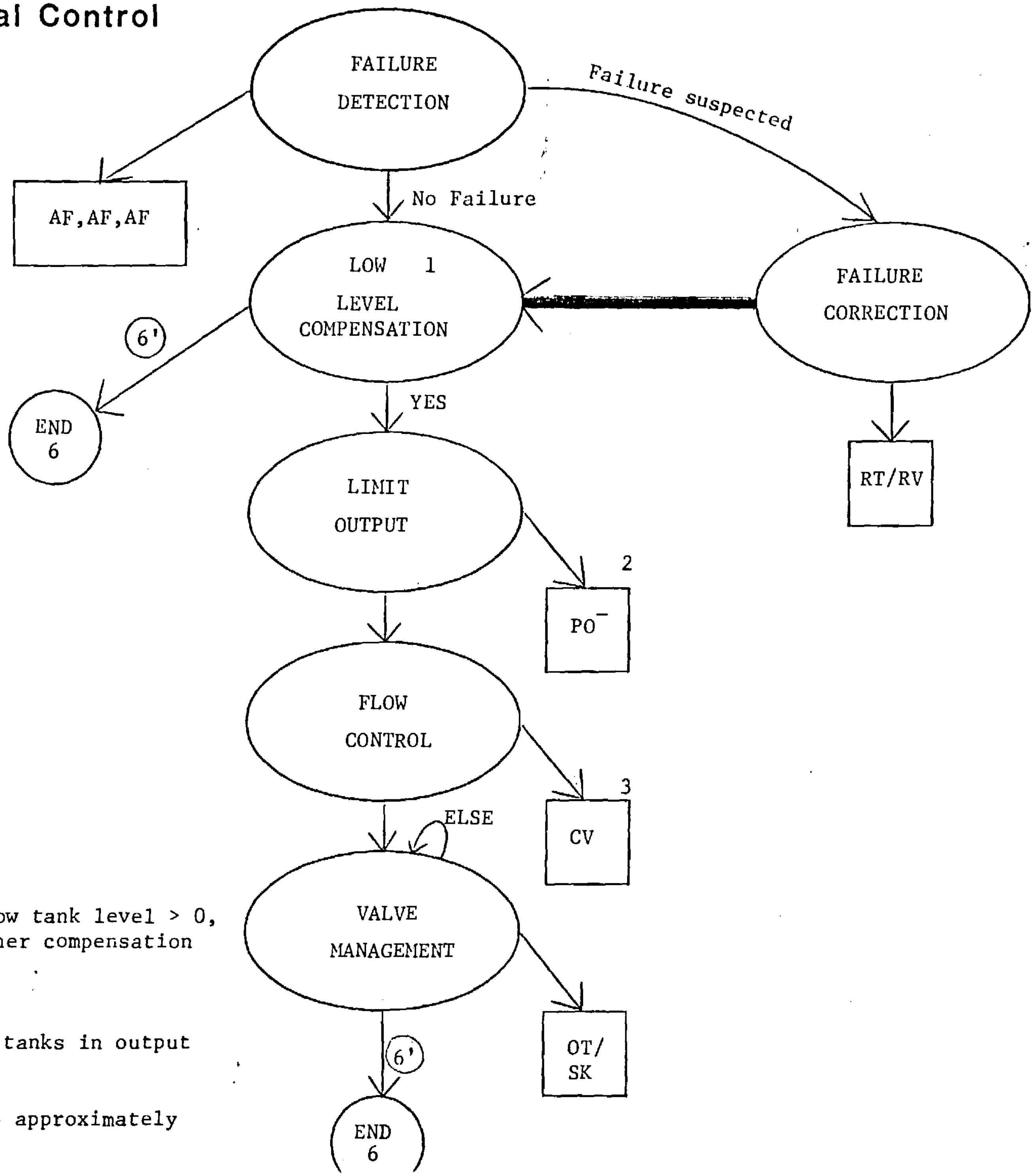


PLANT - Operational Control

Procedure 6:

WC Problem
in Output Column

Figure 10



1: Decision Rule: If low tank level > 0, further compensation

2: PO: = (0 - 50)

3: Close valves to high tanks in output column

6': All tanks in column 3 approximately the same height reduced. These activities continue until the system stabilizes or the repair on the failed component is completed.

Procedure 6 begins with a failure detection and, if necessary, compensation subfunction; then, a decision is made about whether compensation is needed. If not the procedure is ended. If more intervention is required, output is limited, flow to high output tanks is restricted, and tripped valves are opened until tank heights in column 3 stabilize.

Fault Identification and Emergency Management

The final high level control function is fault identification and emergency management (figure 11). This control function is always reached from steady-state management and the transition occurs due either to a suspected or detected fault or to a loss-of-control feeling on the part of the operator.

The system trip subfunction is the least straightforward. The option is invoked when the system is so unstable that the operator feels a total loss of control. It might be argued that this procedure is never really required or justified; a competent operator has less catastrophic procedures available.

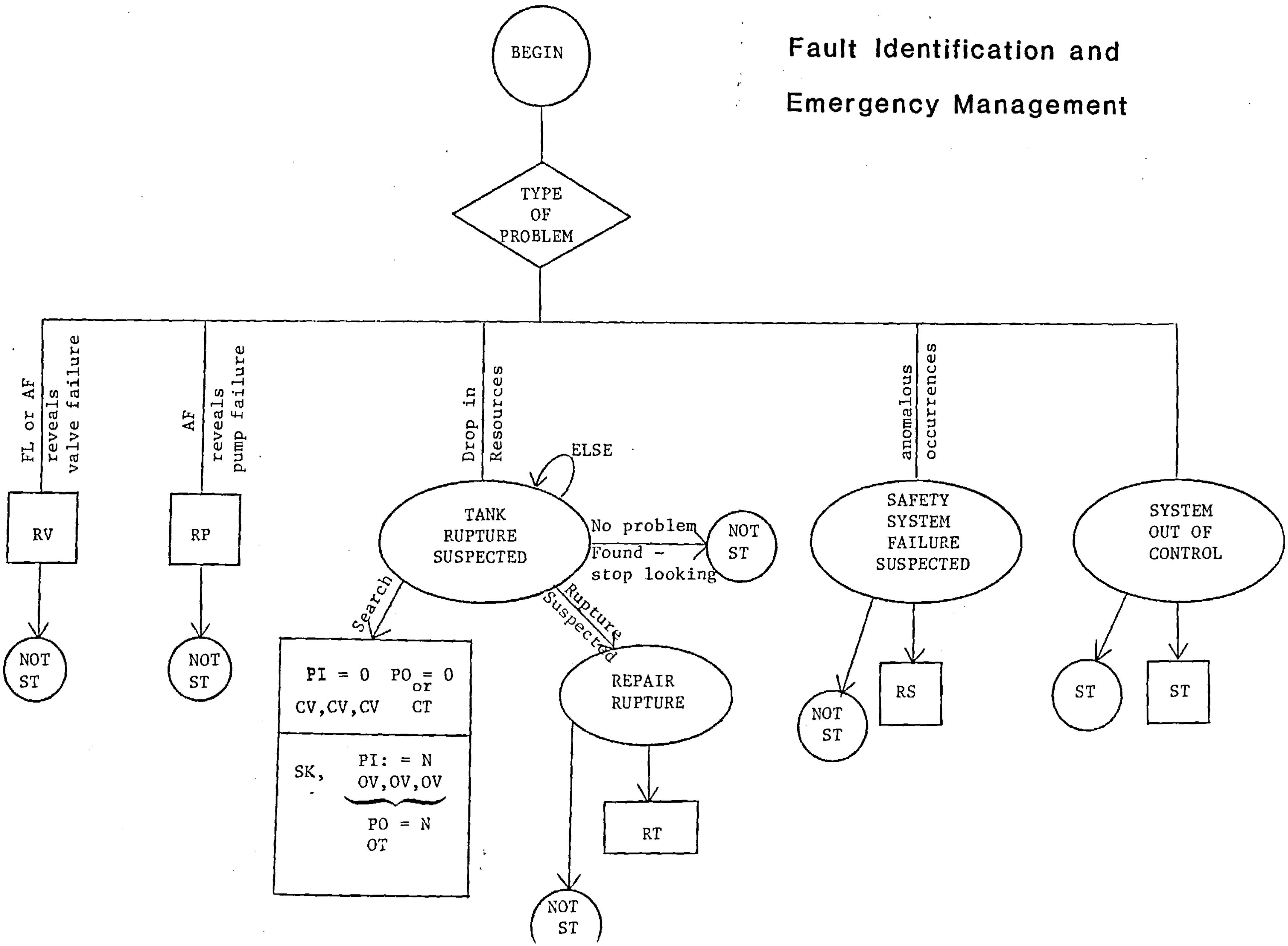
The rest of the subfunctions within the fault identification and emergency management function address requirements for fault detection and diagnosis. Repairs to valves and pumps are straightforward and are likely to occur as a result of routine valve checks conducted in steady-state management.

The procedure to detect and correct a tank rupture is not quite so straightforward. The procedure may be invoked because the operator

Figure 11

PLANT

Fault Identification and Emergency Management



observes an unexplained drop in system resources. The search procedure is laborious and may terminate in detecting a tank rupture, concluding that there is no tank rupture, or stopping an uncompleted search to undertake another control function.

The final fault identification subfunction is the identification and repair of the PLANT safety system. This is a little used procedure and is involved only when the operator has observed a critical number of anomalous system-initiated occurrences.

Uses of the PLANT Discrete Control Model

This model has several potential applications. It can be used either as an analysis tool to aid in understanding operator control performance or as a design tool to create a dynamic human-computer interface to control PLANT.

As an analysis tool, the model can be used to help explain operator control actions given system state. Its normative but nondeterministic form allows a great deal of flexibility in the sequence of control actions that are permitted within a control function or even within a subfunction. The structure is perhaps somewhat more people-oriented than KARL (Knaeuper, 1983), in that procedures have a beginning and an end; and once a procedure is initiated, it will not be preempted or terminated until a completion point has been reached. KARL, exhibiting one of the greatest strengths of computers, meticulously examines all system variables, updates its statistics every iteration, and assesses, given the new state, what it should be doing. As a result KARL could leave a procedure before completion and/or hop from procedure to procedure. People, given cognitive limitations, are much more likely, once beginning a procedure, to continue its steps to conclusion, almost disregarding other symptoms arising in the intervening time. The discrete model has similar characteristics.

Although subject data has not been compared to the discrete control model, a logical next step in the model's validation is to compare it to human performance and evaluate its explanatory power. The model should have a high validity when compared to a "good" operator, i.e., one who had good output and faithfully followed prescribed procedures. The model should also be very helpful at identifying mismatches. The normative nature of the model suggests that these mismatches are likely to be operator mistakes.

As a design tool the discrete control model can be used to design an information display system which selects out, aggregates, or prioritizes system state information to facilitate the operator's control decisions depending on current operator control function and system state. The display system would have individual display pages that are tailored to the needs of operator control functions or subfunctions, as specified by the model. Used in this way, the discrete control model has potential utility as the basis of an on-line interactive decision aid. One strategy would be to let the operator specify the control function or subfunction currently underway, and given that function, the aid could prompt the operator with suggested next steps and activities as well as provide the required pieces of information.

An aid such as this may avoid some of the problems that KARL encountered when used as a human decision aid (Knaeuper and Morris, 1984). By allowing the operator to set the pace and having the program act as an aid rather than an expert, some of the problems of a <u>nagging</u>

on-line aid may be minimized. The interactive nature of an aid based on a discrete control model gives human operators much more control over the human-computer interface. This may be a desirable feature of an aid because as long as the human has the responsibility, s/he probably ought also to have the authority. The nondeterministic, heterarchic nature of discrete control models ensures, and in fact requires, this type of interface. The flexibility built into the interface requires that the human specify where s/he is and what the intent is. Yet the hierarchic structure of the model ensures that once the aid knows where the human wants to be in the control heterarchy, the appropriate information or procedural prompts can be provided.

REFERENCES

1. Knaeuper, A.E. A Model of Human Problem Solving in Dynamic Environments, M.S. Thesis, Center for Man-Machine Systems Research, Georgia Institute of Technology, Report No. 83-3, 1983.

2. Knaeuper, A.E. and Morris, N.M. "A Model-Based Approach for Online Aiding and Training in Process Control," Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, 1984.

3. Miller, R.A. Identification of Finite State Models of a Human Operator, Final Report AFOSR Grant No. 77-3152, The Ohio State University Research Foundationn, 1979.

4. Miller, R.A. "A Systems Approach to Modeling Discrete Control Performance," in W.B. Rouse (ed.), Advance in Man-Machine Systems Research, Vol. II, JAI Press Inc., Greenwich, CO, 1985.

5. Mitchell, C.M. The Design of Computer-Based Integrated Information Displays, Ph.D. Thesis, The Ohio State University, 1980.

6. Morris, N.M. Human Problem Solving in Process Control, Ph.D. Thesis, Center for Man-Machine Systems Research, Georgia Institute of Technology, Report 83-2, 1983.

UNDERSTANDING AND AVOIDING POTENTIAL PROBLEMS IN IMPLEMENTING AUTOMATION

William B. Rouse
Georgia Institute of Technology
Atlanta, Georgia 30332

and

Nancy M. Morris
Search Technology, Inc.
Norcross, Georgia 30092

ABSTRACT

Technology-driven efforts to implement automation often encounter problems due to lack of acceptance or begrudging acceptance by the personnel involved. It is argued in this paper that the level of automation perceived by an individual heavily influences whether or not the automation is accepted by that individual. The factors that appear to affect perceived level of automation are discussed. Issues considered include the impact of automation on the system and the individual, correlates of acceptance, problems and risks of automation, and factors influencing alienation. Based on an understanding of these issues, a set of eight guidelines is proposed as a possible means of avoiding problems in implementing automation.

INTRODUCTION

The impact of technology on society has been debated for at least two hundred years. The initial issues included safety and health of workers, and potential loss of jobs. In recent years, the debate has both broadened and become more focused. It has broadened in the sense that the concerns now include the safety and health of the community as a whole rather than just the workers. It has focused by looking more carefully at the individual and studying how his or her perceptions and behaviors are affected by technology.

Of course, technology has affected individuals for thousands of years. Humans have invented a great variety of tools, devices, and machines. These inventions have made tasks easier, enabled improved performance of tasks, or enabled performance of tasks that would not otherwise be possible.

Initially, people invented tools for their own use. Eventually, people began producing tools to be used by others--and the forerunner of the contemporary hardware store probably emerged. At some point, machines were invented that could perform simple tasks autonomously. Finally, machines emerged that could actually replace people in jobs once thought to be the sole province of humans.

Automation emerged in the transition from technology designed for assisting people to technology designed for replacing people. While the term "automation" is only a few decades old, the concept has existed for many hundreds of years. However, it is only in relatively recent years that automation has come to be viewed as a potentially serious threat to humans' roles in a variety of contexts. This paper explores the nature and potential of this threat.

Exactly what is perceived as threatening? It cannot be the mere existence of automation since most, if not all, people readily accept and value a variety of automatically-controlled devices and machines. Examples of such "everyday automation" include automatically-controlled heating and cooling, self-service elevators, and automatic automobile transmissions. People tend to view these types of automation as very positive, most likely because such automation eliminates tedious or boring tasks, or tasks that distract them from their primary interests.

Automation is viewed quite differently when it impinges on people's primary tasks. In the aviation domain, pilots are concerned that the essence of their jobs will be automated, except in problem situations where they will be expected to "save the day" [1,2,3]. In the area of manufacturing, skilled machinists are justifiably troubled by the fact that automation has reduced their jobs to simply monitoring, tuning, and tending the computers that now control the machining [4,5]. In the office environment, various commentators have noted and expressed some concern about people spending increasing amounts of their time interacting with machines, or interacting with each other via machines, and spending much less time interacting face to face [6,7].

Perhaps people's apprehensions about these trends in automation are simply natural reactions to technological and social change. Perhaps automated airplanes, factories, and offices will eventually come to be viewed as "everyday automation". Alternatively, it seems reasonable to hypothesize that technology and society may be undergoing a qualitative change that does not simply push society along the path of development that has been followed for decades. Instead, this change may bridge a discontinuity and move society to a new path, one that is not understood nearly as well as might be expected. While this hypothesis is not explicitly pursued in this paper, much of the discussion relates to this issue.

FUNCTIONS OF AUTOMATION

It is rather pointless to debate whether automation, or technology in general, is good or bad, a blessing or a curse [8]. As discussed earlier, some types of automation appear to be accepted readily and valued, while other types cause more apprehension and even fear. Thus, no blanket acceptance or rejection is possible.

It seems reasonable to assert that what discriminates acceptance from rejection is perceived level of automation. The concept of level of automation has been elaborated by Sheridan [9] and utilized by the Air Force Studies Board [2]. Sheridan's ten levels range from manual control on the low end to total automation on the high end. In order to understand how people might perceive these levels, Dieterly [10] has argued that one should consider the functions which are automated and people's perceptions of these functions. Based on recent work on the nature of decision making tasks [11], it seems reasonable to conclude that each of Sheridan's ten levels (except manual control) involve automation of one or more of the following functions:

To appear in Proceedings of the 1985 International Conference on Systems, Man, and Cybernetics, Tuscon, AZ, November 12-15, 1985.

1. Synthesis = generation of alternatives
2. Analysis = evaluation of alternatives
3. Decision = selection among alternatives
4. Control = implementation of alternatives
5. Monitoring = observation of results

Before considering the attributes of these functions that might affect perceived level of automation, it is useful to discuss examples of each function.

Examples

The most common functions of automation are monitoring and control. Everyday examples of monitoring include smoke detectors, fire alarms, and burglar alarms. Examples of automated monitoring and control include thermostats, automatic transmissions, and autopilots.

Automated decision making, or selection among alternatives, is difficult to find except in situations where time constraints preclude human reactions (e.g., "scram" of a nuclear power plant). Further, even for the rare examples that do exist, one would have difficulty claiming that the computer is exercising "judgement". In fact, it would be reasonable to claim that such pre-programmed selections do not really represent decision making on the part of the computer.

Automated synthesis and analysis have emerged in the form of "expert systems" for medical diagnosis, chemical analysis, and geological evaluation. These systems are automated in the sense that they produce and analyze alternatives. However, in general, their autonomy is extremely limited because their typical users are very knowledgeable professionals who can readily evaluate and pass judgement on the systems' results.

Implications

From the above examples, it seems reasonable to conclude that humans will primarily discriminate among the five functions in terms of level of discretion allowed or required. In the context of this paper, discretion can be defined in terms of opportunities for exercising creativity and judgement. While the five functions do not map neatly to scales of creativity and judgement, it is clear that synthesis usually involves more of these abilities than are involved in the other four functions. Similarly, analysis and decision usually involve more judgement, and occasionally creativity, than do control and monitoring.

While level of discretion, or extent of opportunities for exercising creativity and judgement, appears to discriminate among Sheridan's ten levels of automation in a behaviorally meaningful manner, this refinement is not sufficient to explain differences between perceived and actual levels of automation. It is quite possible for a device to be highly automated but not perceived as such because its level of discretion is not high relative to the interests of the perceiver. Therefore, perceived level of automation will be high to the extent that the functions which are automated preclude the observer from desired levels of discretion relative to his or her task objectives. Put simply, most people would gladly give their automobile transmissions discretion about when to shift, but would not like to lose discretion about where the automobile goes.

To summarize, it seems reasonable to hypothesize that perceived level of automation is directly related to the level of desired discretion that is automated. If this hypothesis proves to be reasonable, then acceptance of automation can be increased by manipulating either the discretion of the automation and/or the desired discretion of the personnel involved. While this strategy may appear straightforward, it actually requires an understanding of a variety of factors, many of which are discussed in the following sections.

THE IMPACT OF AUTOMATION

In order to discuss humans' acceptance of automation it is important to consider first what they are being asked to accept. More specifically, what are the impacts of automation that are to be accepted? This section elaborates upon these impacts in terms of both the system and the human.

Impact on the System

Interest in automation is usually based on the potential benefits to be realized. Improved performance/productivity, safety, and economy are often anticipated, regardless of whether the domain is aviation [1,2,3], manufacturing [4,5], or the office [6,7]. The basis for these expectations is often the possibility of more flexible use of equipment and space [3,4,5], as well as more accurate and reliable control/production.

A less obvious, but potentially more far-reaching, impact of automation is increased managerial or organizational control [5,12]. Computer-based, integrated air traffic control systems and computer-based, flexible manufacturing systems not only have the potential for improved performance/productivity, they also provide ready access to performance/productivity metrics for each pilot, machinist, typist, etc. The availability of such a breadth and depth of information offers several obvious potential benefits and a few less-obvious potential dangers.

From a benefits point of view, there is opportunity for a management control strategy that approaches global optimality, rather than suboptimality due to local strategies that ignore system-wide objectives. Ineffective and inefficient aspects of system operations can be more quickly identified and resources dynamically reallocated. Thus, for example, rather than have idle resources, component parts can be produced "just in time" for humans (or robots) to assemble them into a larger system [5].

Beyond the obvious pressures that centralized control can present for humans who may have difficulty with the lock-step pace necessary for the "just in time" concept to work, there are two potential problems that are much more pervasive. The least subtle of these is the fact that individual workers can be monitored, perhaps online in real time, by a centralized system. This might result, for example, in a word processor informing a user that his or her productivity (e.g., pages per hour) is off or number of grammatical and spelling errors up this morning. Further, of course, the supervisor of this individual could access these metrics and perhaps "attach" to his or her terminal to provide pep talks when necessary!

This rather obvious danger has received much attention by many commentators and is not pursued further in this paper. A much more subtle danger of centralized control is the strong possibility that globally-optimal policies might not be comprehensible in local contexts. For example, a reallocation of resources that makes wonderful sense from the point of view of the total system may seem very counterproductive to those whose purview is limited to portions of the system that will be net losers of resources. This could be particularly problematic if the "extra" resources are due to improved human (as opposed to technological) productivity.

Thus, what emerges is a situation where humans in any one part of the system cannot exert substantive control over that part of the system because they lack knowledge of the objectives and operations of the total system. As a result, they may be responsible for implementing policies that they inherently cannot fully explain. This possibility can severely undercut acceptance of automation and perhaps lead to alienation, two topics that are discussed in some detail later.

The phenomenon of lower-level units of large organizations not fully comprehending policies developed by a centralized management is certainly far from new. However, the potential problems of such situations are substantially aggravated by computer-based information systems in conjunction with high levels of automation. This is due to the fact that the "agent" of the less than fully comprehensible policies becomes technological rather than human. There is no longer any recourse to a human supervisor or manager who understands the basis of the misunderstood management policies.

Such a situation is due in part to technological trends toward large-scale centralized systems, and in part to an assumption that humans need not fully understand the objectives of the systems in which they work. This assumption might be tenable if these systems were fail-safe. However, in all cases of automation to date, humans remain as the ultimate backup system [1,13,14]. For example, when the automation fails, due to either a design limitation or a hardware/software problem, humans have to intervene and assure that normal operations are recovered. It is unclear how humans are supposed to fulfill these responsibilities if they do not understand those overall objectives of system operations that are relevant to establishing priorities in such failure situations.

To summarize, the system-wide impact of automation includes potential improvements in performance/productivity, safety, and economy. Increased organizational control also offers possibilities for more global optimization of operations. These potential benefits have associated risks. Global optimization and control can lead to local lack of understanding, as well as possible mistrust and alienation, particularly if centralized, computer-based monitoring of worker performance/productivity is an important element of overall managerial control policies.

Impact on the Individual

It is important to avoid limiting discussion to the system-wide impact of automation. Some of the more compelling effects of automation include the impact on the individual's tasks and behaviors. This is due to people's rather natural tendency to perceive change in terms of how it impacts them rather than in more conceptual or abstract terms.

Upon reviewing the projections of various commentators on this issue [4,5,7,9], is seems reasonable to generalize several trends across various domains of application. With increased automation, humans in general will be:

1. Performing tasks less; watching and listening more,

2. Walking around and communicating face to face less; key pressing more,

3. Exercising much less discretion; coping with much more complexity.

These trends may make sense if the only concern is machine-like productivity in normal, routine operations. A key issue is whether or not it is reasonable for this to be the only concern.

There are two reasons why such a perspective is very short-sighted. First, despite substantial investments and designers' skill and good intentions, not all operations are normal and routine. Unfamiliar and failure situations emerge and, as emphasized earlier, humans are expected to intervene and take control. Some of the above trends are inappropriate and probably counterproductive relative to humans filling their backup roles.

The second reason is more subtle. Humans are social animals. Whether the environment is an aircraft cockpit, a factory, or an office of engineering designers, people need to interact with each other, and they value this interaction greatly. Indeed, face-to-face interaction, including exchanging of job-related stories, is an important and usually underestimated means of gaining information. Tidbits and nuances learned in this way can have profound effects on people's abilities to solve novel problems that later emerge. The trends noted above will substantially decrease this type of learning.

A discussion of the impact of automation on the individual would be incomplete if workload was not mentioned. One of the frequent justifications for pursuing automation is a need to reduce mental and physical demands on the personnel in the system. While some arguments can be made for automation having reduced physical workload, most results to date indicate that mental workload is usually increased [3,5]. This is due in part to designers' penchant for increasing system capabilities in the process of automating, and in part to automation requiring set-up, monitoring, and intervention activities that did not previously exist. Thus, the intended benefits for the humans in the system are seldom realized.

ACCEPTANCE OF AUTOMATION

While the previous section emphasized potential problems with automation, this does not mean that potential benefits are insufficient to outweigh these costs. In fact, the choices usually made by the techologically-oriented portions of society would seem to indicate that the benefits are perceived as outweighing the costs. Of course, the decision makers in question are not the same individuals who have to pay the types of cost outlined earlier in this paper.

The individuals paying these costs are likely to have to deal with automation as a fait accompli. The issue for them, therefore, is not one of choosing automation but whether or not to accept it. Automation is accepted to the extent that it is favorably received and willingly used as intended. A lack of acceptance does not imply that the personnel involved strike or quit, but that they begrudgingly employ the automation if at all.

Acceptance of automation has been found to increase with automation of tasks that humans cannot perform, distracting tasks, and tedious tasks [2]. It is, obviously, also necessary that this automation be reliable. Acceptance also increases as the personnel involved gain more experience with automation, and is greater for personnel who have relatively more status, responsibility, and authority [16].

Acceptance decreases when the functions automated are perceived as involving decision making and when the costs of failure are high [16]. Acceptance also decreases when the personnel involved have no discretion in selecting modes of automation and intervening if they feel it to be necessary [2]. Finally, a variety of automation-specific problems and risks can lead to decreased acceptance. These problems and risks are discussed in the following section.

Summarizing, acceptance increases when automation reliably performs tasks that people cannot do or would rather not do. Acceptance decreases when the tasks which are automated are viewed by the humans in question as uniquely their province. Acceptance also decreases when the humans in question view themselves as having no discretion in the use of the automation.

PROBLEMS AND RISKS

As noted above, acceptance can also be undermined by a variety of automation-specific problems and risks. Wiener and Curry [1], Sheridan and his colleagues [17], and other commentators have elaborated upon these considerations. This section only presents those issues necessary to the line of reasoning being developed in this paper.

The most obvious problem is failure of the automation itself. As earlier discussion has emphasized, many of the potential difficulties for humans are due to the fact that they have to be prepared to intervene and assure recovery. People's concern that they will be ill-prepared for this role could certainly affect acceptance.

Humans errors in setting up automation have led to some well-documented aviation incidents and accidents [1,3]. Automation usually does what it is told, even if its instructions are inappropriate. If humans do not have the means and motivation to check what is happening, automation can "blindly" fly into off-limits territory or into the side of a mountain! Humans' concern that they might supply inappropriate inputs could lead to their avoiding automation or duplicating its efforts as a means of checking.

Humans can also be misled by automation. False alarms and missed events by automated caution, warning, and alarm systems can lead humans to be distracted by spurious events, and to be unaware of important events, respectively. Fault-tolerant control systems can progressively compensate for an increasingly degraded system, and avoid "bothering" the personnel involved until the situation is almost hopeless. When humans are misled in such ways, they tend to avoid using and/or depending on the automation.

The types of problem discussed thus far would seem to be amenable to solution by improved and more sophisticated technology. There are other types of problem, however, that do not seem quite so straightforward. One of these problems is humans' consistently poor performance in monitoring complex systems. When humans' primary role is simply watching, they have a very difficult time remaining vigilant and interpreting what they see. As a result, they are ill-prepared to detect that intervention is necessary and, subsequently, act appropriately.

Beyond vigilance and interpretation problems, humans have a great deal of difficulty in acquiring and retaining skills that they hardly ever use. With the current state of technology, automation is typically monitored by pilots and machinists who have developed and refined their respective skills over many years. As a result, they know what should happen and how it should be done. This enables them to monitor, tune, and intervene effectively. With increasing automation, how will such skilled personnel be developed in the future? In fact, many current pilots and machinists complain about loss of skills due to lack of use.

Perhaps the biggest risk is that people will become alienated by automation in particular and technology in general. If this happens to a large enough portion of society, then either technological change will be substantially slowed, or a technological elite will emerge which is inconsistent with society's usual egalitarian orientation. To devise possible means of avoiding these results, the factors that appear to lead to alienation must be explored.

FACTORS INFLUENCING ALIENATION

There exist possibilities for alienation throughout the process of technological change. Prior to implementation of an increased level of automation, there may be misunderstanding of the impending change [10]. This can lead to perceived inferiority and threatened obsolesence [9], perhaps due to people's inability to view their roles in the system as dynamic [10].

Once implementation occurs, humans may feel increasingly remote from their tasks [9] and, consequently, feel a loss of value of their sensory/motor skills [1,5,9,17]. It is also likely that there will be decreased interaction with other humans, and more interaction with computer input devices [7]. In general, humans may feel a lack of discretion or freedom, and an overall loss of control [13].

On a longer term, as people gain experience with the system, acceptance may increase as noted earlier. However, also possible are several more pervasive aspects of alienation. The personnel in the system may place too much trust in the automation, endowing it with almost mystical abilities [9,17]. This can lead to diffusion and abandonment of responsibility [13,17], which presents obvious problems since the automation is unlikely to have any social or legal responsibility for its behaviors. Finally, any or all of the factors outlined in this section can lead to decreased job satisfaction, prestige, and self-concept [1,17].

It is easy to see how alienation can emerge. A lack of understanding of upcoming change, anticipated obsolesence, and loss of control would alienate many people, regardless of whether automation is involved or not. If the risks of alienation are to be minimized, and the chances of acceptance maximized, the development and implementation of technology must be approached in a different way.

SHIFTING THE EMPHASIS

A prerequisite to developing an appropriate approach to automation is explicit recognition that the goal is to support the personnel involved so that they may achieve system objectives of performance/productivity, safety, and economy. From this perspective, the use of automation technology is a means rather than an end. Thus, automation efforts should be objectives-driven instead of technology-driven.

Assuming that an increased level of automation is the best means to the above goal, how can user acceptance be assured? The earlier discussions in this paper would seem to indicate that acceptance is more likely to be high if the level of automation is not perceived as excessive. User acceptance can, therefore, be affected by manipulating perceived level of automation. The following guidelines seem to provide a reasonable synthesis of the variety of issues and perspectives discussed in this paper:

1. To the extent possible, only automate system functions that personnel in the system feel should be automated (i.e., those for which they are willing to lose discretion),

2. To the extent necessary, particularly if the first guideline cannot be followed, increase the level and number of functions for which personnel are responsible so that they will be willing to delegate lower-level functions (i.e., expand the scope of their discretion),

3. Assure that the level and number of functions allocated to each person or type of personnel form a coherent set of responsibilities, with an overall level of discretion consistent with the abilities and inclinations of the personnel,

4. Avoid automating functions when the anticipated level of performance is likely to result in regular intervention on the part of the personnel involved (i.e., assure that discretion once delegated need not be reassumed).

These general guidelines reflect what might be termed necessary conditions for personnel to accept increased automation. However, these conditions are not sufficient. Recalling the discussion of alienation, it is important to consider how personnel will anticipate a planned increase in automation. From this point of view, the following additional guidelines appear relevant:

5. Assure that all personnel involved are aware of the objectives of the automation effort and what their roles will be after the change,

6. Provide training that assists personnel in gaining any newly required skills and helps them to internalize the personal value of having these skills,

7. Involve personnel in planning and implementing the changes from both a system-wide and individual perspective,

8. Assure that personnel understand both the abilities and limitations of the increased automation, know how to monitor and intervene appropriately, and retain clear feelings of still being responsible for system operations.

The eight guidelines offered in this concluding section are meant as a starting point, perhaps as a next step from the Air Force Studies Board's guidelines [2]. While the empirical basis of these guidelines is far from rigorous, it is not clear that they can be "validated" in any traditional sense. Nevertheless, they are proposed with a primary goal of fostering a user-oriented approach to automation, an approach that emphasizes supporting humans rather than replacing them.

ACKNOWLEDGMENT

This research was supported by the National Aeronautics and Space Administration under Ames Grant NAG 2-123. Dr. Everett Palmer is Technical Monitor for this grant.

REFERENCES

[1] E.L. Wiener and R.E. Curry, "Flight-deck automation: promises and problems," Ergonomics, vol. 23, pp. 995-1011, 1980.

[2] Air Force Studies Board, Automation in Combat Aircraft. Washington, DC: National Academy Press, 1982.

[3] E.L. Weiner, "Beyond the sterile cockpit," Human Factors, vol. 27, pp. 75-90, 1985.

[4] M. Blumenthal and J. Dray, "The automated factory: vision and reality," Technology Review, pp. 29-37, January 1985.

[5] H. Shaiken, "The automated factory: the view from the shop floor," Technology Review, pp. 17-27, January 1985.

[6] M.G. Helander, "Emerging office automation systems," Human Factors, vol. 27, pp. 3-20, 1985.

[7] H.M. Parsons, "Automation and the individual: comprehensive and comparative views," Human Factors, vol. 27, pp. 99-112, 1985.

[8] G.S. Moschytz, "The challenge of modern technology: a blessing or a curse?" IEEE Technology and Society Magazine, pp. 3-26, March 1983.

[9] T.B. Sheridan, "Computer control and human alienation," Technology Review, pp. 61-73, October 1980.

[10] D.L. Dieterly, "Automation in organizations: eternal conflict," NASA Ames Research Center, Moffett Field, CA, Tech. Mem. 81290, May 1981.

[11] W.B. Rouse and S.H. Rouse, "A framework for research on adaptive decision aids," Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, OH, Tech. Rept. No. AFAMRL-TR-83-082, October 1983.

[12] K.B. DeGreene, "Man-computer interrelationships," in K.B. DeGreene, (Ed.), Systems Psychology. New York: McGraw-Hill, 1970.

[13] J.G. Wohl, "Information automation and the Apollo program: a retrospective," IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-12, pp. 469-478, 1982.

[14] J. Sharit, "Supervisory control of a flexible manufacturing system," Human Factors, vol. 27, pp. 47-59, 1985.

[15] E. Edwards, "Automation in civil transport aircraft," Applied Ergonomics, vol. 8, pp. 194-198, 1977.

[16] H.E. Price, "The allocation of functions in systems," Human Factors, vol. 27, pp. 33-45, 1985.

[17] T.B. Sheridan, T. Vamos, and S. Aida, "Adapting automation to man, culture, and society," Automatica, vol. 19, 1983.

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

Wan C. Yoon and John M. Hammer

Center for Man-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

To appear in Proceedings of the 1985 International Conference on Systems, Man, and Cybernetics, Tucson, AZ, November 12-15, 1985.

An aid is proposed for the operator who must deal with a novel failure. A novel failure is one that is not covered by the operator's training or procedures or by an expert system (if present). The aid contains a disaggregated model of the system for reasoning causally about the system. It is to work in parallel with the human and interact at various levels of control. It is designed specifically to mitigate some human suboptimalities and biases during decision making.

Introduction

In highly automated systems, the human operator is primarily a monitor and supervisor. A very important monitoring function is diagnosing equipment faults, an increasingly difficult task in automated systems. The current approach to fault diagnosis is to train the operator to deal with common faults, usually with training and specific written procedures. This approach has obvious limitations when the fault is not covered by training or procedures.

Recently, there has been much interest in supporting the human operator via expert systems for diagnosis. To be sure, this approach will improve the system performance on relatively common failures. Generally speaking, however, the expert systems are subject to the same limitations as training and procedures: the designer usually has to anticipate the failure for the expert system to solve it correctly.

The approach used in designing this aid is to provide it with a disaggregated model of the equipment and to let it work to solve the problem in parallel with the human. It specifically aims to mitigate certain human suboptimalities during decision making.

The disaggregated model will most likely be a qualitative model [2]. It will provide the aid with some ability to diagnose the problem by reasoning causally about the system. It is claimed that a qualitative model represents the system in the same terms as does the human [4] [3]. Furthermore, it should be robust. Davis [1] has argued that systems that reason from an understanding of causality of the device can cover a wider range of faults than the traditional expert systems using collections of rules based on experience.

Sequential interaction is the norm for most human-computer systems. It would be advantageous, however, if the human and computer both perform similar tasks in parallel, with some degree of independence from each other [8]. Then each, especially the human, can freely pursue his own strategy without being forced to change planned actions except when information is needed from the other diagnostician. This parallelism causes a wide range of interaction levels which we will discuss later in this paper.

Mitigation of decision making biases is an important part of the aid. These biases can cause serious performance problems. Experimentally, we hope to demonstrate that the aid improves human diagnosis on novel failures.

The Task and The Interface

The Task

There is a system of components (e.g., pumps, valves, switches) and connections (e.g., wires, pipes). There are flows (e.g., current, heat) between components through connections. The operator's task is to find the states of the components (e.g., on, off) and flows (e.g., low, high) that explain the observed symptoms. Some states are known (in particular, the symptoms), and some must be inferred.

The behavior of the system is governed by component rules that are selected by component state [2]. For example, a switch might be defined as follows. If the switch is _closed_, then its input and output currents are equal. If the switch is _open_, then its input and output currents are zero. _Closed_ and _open_ are states. It is also possible to imagine other states which represent faulty behavior. A legal explanation is an assignment of states to components and flows that are consistent. To be consistent, the flow state must be equal at both ends of a connection. It should be obvious that an explanation could be found by combinatorially searching through all possible component state assignments. The operator's task is to find this explanation. We do not consider fault correction or compensation here, even though these problems are important. Presumably, a similar kind of search for an acceptable state would work.

The Interface

The interface will consist of a display with windows for the schematic, hypotheses, assumptions, and interaction. The schematic window displays a schematic of the system. It will show the components, connections, and the states assigned to them (if known). A system too large to fit on a screen will be hierarchically displayed. The operator will be able to zoom in and out to the appropriate level of detail. The schematic will represent the current state of reasoning about the equipment.

The hypotheses window will display the current set of hypotheses about what could cause the system to fail. These hypotheses take the form of state assignments to components or flows (e.g., switch3 : closed). The list is created first by the aid. The human may examine and modify the list.

The assumption window will display assumptions made by the operator. An assumption is simply an assumed state (e.g., switch 13: on). Assumptions are needed because some equipment is configured such that it is impossible to deduce states without assuming something. If the window cannot display all the hypotheses simultaneously, they may be scrolled within the window. Also attached to a hypotheses will be a plausibility indicator. Currently displayed hypothesis will have their plausibility updated whenever new evidence requires it. The operator may also request the list to be sorted and redisplayed in order of plausibility.

Interaction will be initiated in the interaction window. The actions available to the operator would include:

(1) Making assumptions.

(2) Asking the aid to search for a consistent set of states.

(3) Changing the level of the schematic display.

(4) Asking the aid to explain a chain of reasoning of how a particular state was determined.

(5) Manipulating the hypotheses list.

(6) Asking what-if questions.

An additional possibility not heretofore mentioned is that of creating new states for components. Our intent is to make the component models complete, but there is always the possibility that a component could be in a state that is not part of the model. The operator will be allowed to create a new state with its own, new behavior.

Mitigating Suboptimalities

A goal for this project is not only to aid human problem solving but also to mitigate human suboptimalities or biases in decision making [10]. In this section we will enumerate these suboptimalities. The suboptimalities break down into two general categories: those that can be mitigated and those that cannot. Our definition of mitigation is that the aid takes some specific action to reduce or eliminate the suboptimality before it occurs. This action must go beyond the aid's simply reasoning without bias about the problem. An example of a suboptimality that can be mitigated is limited short-term memory. It can be aided by using a display as a form of memory. An example of a suboptimality that cannot be mitigated is ignoring base rate probabilities. (An example of this would be an initial diagnosis of an extremely rare disease that covered the symptoms exactly). There is little that the aid can do directly to mitigate this except to reason correctly itself. More emphasis will be placed on suboptimalities that can be mitigated in this section.

Suboptimalities That Can Be Mitigated

An incorrect mental model is an operator's model of the system that does not predict the actual system operation under all conditions. Many authorities agree that operators use models to make such predictions, especially under novel circumstances [4] [7]. Our ability to identify and understand limitations in mental models is itself not well understood [9]. The compensation for an incorrect mental model will be the aid's (presumably) more complete and more correct conceptual model. Furthermore, the operator will be able to examine this model at the component level and ask for explanations of a chain of reasoning with the model. One requirement of the model is that its interface representation be compatible with the human's.

Anchoring, confirmation bias, and cognitive tunnel vision are names for phenomena that will be treated in the same way. Humans tend to stay with an initial hypothesis, ignore evidence that disconfirms it, and avoid exploring alternative hypotheses. To mitigate, the aid will list the most plausible hypotheses. When the human anchors to a relatively implausible hypothesis, there will be a discrepancy. Some hypotheses will be ranked far superior to the anchored one. This discrepancy would presumably cause the human to drop the anchored hypothesis.

Another compensating action the aid will take is to undo assumptions. Suppose the operator makes an assumption about the state of some pump, which then leads to a solution. Before allowing the operator to accept this solution as the only one, the aid will undo the assumptions and suggest a search for others.

Humans have limited ability to entertain more than three or four hypotheses or to generate a full set of hypotheses. To mitigate, the aid will display hypotheses in a window and maintain a full set with plausibilities. The operator will be able to interact with this list.

Humans tend to use the more salient data (i.e., perceptually more prominent) in decision making. The aid will not be susceptible to this bias, of course. The aid could mitigate the human's bias by displaying saliently the most diagnostic information. This requires that the aid be able to choose the most diagnostic information.

As mentioned earlier, humans have limited short-term memory. To mitigate, the display will augment memory. The current system state, the set of feasible hypotheses, and any current assumptions will be displayed.

Suboptimalities That Cannot Be Mitigated

For the sake of completeness we list and define here the suboptimalities that cannot be mitigated. As stated earlier, the only remedy for these seems to be to keep them out of the aid's reasoning. Then the disagreement between the aid's results and the human's results may be fed back either by continuous display or interrupting advice. It may be, however, that we have missed some way of actively mitigating them. The suboptimalities are:

(1) Overestimation of the strength of causal relationships.

(2) Representativeness. A diagnosis that matches the symptoms may be chosen even though its a priori probability is relatively low.

(3) Availability. If a hypothesis is easily recalled from memory, it may be used.

(4) Elimination by aspects. If a large number of cues are relevant to a decision, the human may reduce the number of choices by eliminating all that do not qualify on a few, most important cues.

(5) Revising odds over time. Relative to the optimal Bayesian model, humans are conservative in adjusting odds over time as new data become available.

(6) Reasoning difficult when opposite to encoding. The normal operation of a system is most likely encoded as cause to effect (i.e., source to destination). Fault diagnosis requires reasoning from effect to cause. Reasoning opposite to encoding is more difficult than reasoning in the direction of encoding.

(7) Insensitivity to data reliability. When faced with data of unequal reliability, humans tend to treat all data as equally reliable.

(8) Sampling data. The human's sampling of data is biased toward the cheap and unreliable.

The results of these suboptimalities, however, can be detected in the forms of an overlooked hypothesis, an overlooked datum, or poor evaluation of plausibilities. The aid will try to alert the operator to those faults whenever they are apparent.

Suboptimalities of the Aid

The suboptimalities of the aid are also of interest. Since the aid has not yet been completed, we do not know how it will perform. We suspect the following:

(1) The aid may be slow. In this case, some of these computations might be done in the background while the human is thinking and interacting with the computer.

(2) The aid will not be able to innovate. It will be bound by the model.

(3) The aid will not be able to make inductive leaps.

(4) The aid's explanations may be so long as to be useless.

Levels of Interaction

Throughout the preceding discussion we have been silent on the issue of dialogue control. Because the aid has some reasoning abilities about the system and is to intervene during some human suboptimalities, this issue is important. This section explains a philosophy of interaction.

Our aid is designed to work in parallel with the human. By working in parallel. we mean that the aid and the operator both work toward solving the diagnosis problem [8]. Each may have a hypothesis that it regards as most plausible and can take actions to try to determine if that hypothesis is true. Since the aid should at times work independently. the degree of intrusiveness of one problem solver (i.e., the human or the computer) toward another in interaction could have a wide range. For instance, the human could instruct the aid to do a subtask the human finds taxing or error-prone. In another extreme, the computer could pace the problem solving while requesting the human to follow instructions. At an intermediate level, advice and other forms of information exchange are also possible. Such a wide range of interaction levels is possible only when the aid can work in parallel on the problem. This in turn requires the aid to have some diagnostic reasoning ability.

Controlling the interaction levels may be viewed as giving the general scheme of dialogue control. In the following section we stratify into five levels the ways in which the human and computer interact.

In the <u>human-directed level</u>, the computer will respond to the operator's request to perform a subtask or to answer a question. Some subtasks may require a simple data retrieval and others may require highly sophisticated reasoning. They are categorized, however, into the same level since the computer's tasks are initiated by the human and the results are interpreted by the human.

In the <u>human-suggest level</u> of interaction, the human may tune the computer's processing by controlling its intermediate results. For example, the human may constrain the computer's attention by rejecting or reranking some computer-generated hypotheses. Although adjusted by the human, the computer will generate and evaluate hypotheses with its own strategy.

In the middle is the <u>cooperation level</u> in which each problem solver works independently except that one can voluntarily take the other's intermediate results as input. The plausibility indicators, for example, may and may not affect the human's procedure, depending on whether he wants to refer them or not at the point of time. Explicit dialogue is minimal at this level. Each problem solver will seek to evaluate hypotheses more or less independently of the other.

The computer can advise the human in the <u>computer-suggest level</u>. When it is apparent that the human has missed an important point, the computer should interrupt the human with appropriate advice. In response to this, the human will need to modify his problem solving. For example, if the human is investigating hypotheses of low plausibilities while there exists one that the computer thinks is highly plausible, the computer can suggest the hypothesis.

The <u>computer-directed level</u> is a counterpart to the human-directed level. The computer will guide the problem solving by assigning subtasks to the human. The results will also be taken and interpreted by the computer.

Several points need to be discussed in connection with the interaction levels. In a session of problem solving, the level is not necessarily restricted to one or two of the aboves. The level is dynamic and will be controlled by both the human and the computer. The level, unless explicitly maintained elsewhere, will tend towards the center (i.e., the cooperation level), in which the problem solving will be more independent.

It is preferable for the interaction to stay as much as possible in the cooperation level for two reasons. The benefit of parallelism will be maximized at this level. Also, information exchange will be very efficient, if it is possible in this level, in that it is implicit, quick and does not disturb the human's problem solving procedure. Thus, information to be displayed in the cooperation level should be selected carefully to match the needs of the human. The list of highly plausible hypotheses is an example of such information. If it is not found in the display of cooperation level, the human should initiate the human-directed level interaction (i.e., requesting the possible hypotheses and their plausibilities) whenever he needs an aid to select the next hypothesis to be investigated.

Choosing between directed and suggest levels when the computer is to intervene is a technical problem to be solved. The computer should consider the seriousness of possible consequences of a detected bias to determine the interaction level. For the human, if both levels are possible, the human-suggest level may be safer to use than the human-guided level. At the former level, the computer will continue to work and thus give the human some intermediate results to which he can compare his own results.

For example, the human may suspect several hypotheses which are not seriously considered by the computer. At the human-directed level he can examine these hypotheses one by one using the aid's facilities such as what-if inquiries. Alternatively, he can forcibly assign high ranks to those hypotheses so that their evaluation by the computer will continuously be reported on the display. In this way, the computer will give more feedback to human from its independent reasoning.

Conclusion

An aid for novel fault diagnosis is proposed. To the best of our knowledge, this aid is unique in the following ways. It attempts to aid the operator during novel fault diagnosis while most aids are designed for routine problems. It contains a disaggregated model which presumably corresponds to the human's representation of the system. It is designed to work in parallel with the human at changing levels of interaction. It specifically attempts to mitigate a number of human decision making suboptimalities during fault diagnosis.

Acknowledgment

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

References

[1] R. Davis, "Reasoning from first principles in electronic troubleshooting." Int'l. J. Man-Machine Studies, vol. 19, pp. 403-423, 1983.

[2] J. de Kleer and J. S. Brown, "Assumptions and ambiguities in mechanistic mental models," In D. Gentner and A.L. Stevens (Eds.), Mental Models. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

[3] K. Forbus and A. Stevens, "Using cumulative simulation to generate explanations," Bolt Beranek and Newman Inc., Report No. 4490, 1981.

[4] D. Gentner and A. L. Stevens (Eds.), Mental Models. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

[5] J. A. Landeweerd, "Internal presentation of a process, fault diagnosis and fault correction," Ergonomics, vol. 22. No. 12, pp. 1343-1351. 1979.

[6] J. Rasmussen. "The human as a systems component," In H.T. Smith and T.R.G. Green (Eds.), Human interaction with computers. New York: Academic Press, 1980.

[7] J. Rasmussen, "Skills, rules, and knowledge: signals, signs,

and symbols, and other distinctions in human performance models," _IEEE Trans. Systems, Man, and Cybernetics_, SMC-13, No. 3, pp. 257-266, 1983.

[8] M. E. Revesman and J. S. Greenstein, "Human/computer interaction using a model of human decision making," _Proceedings of 1982 IEEE Int'l Conference on Cybernetics and Society_. Seattle, October. pp. 449-453.

[9] W. B. Rouse and N. M. Morris. _On looking into the black box: prospects and limits in the search for mental models_. (Report No. 85-2). Atlanta, GA: Center for Man-Machine Systems Research, May 1985.

[10] C. D. Wickens, _Engineering psychology and human performance_. Columbus, OH: Charles E. Merril Pub. Co., 1984.

SIGNIFICANCE TESTING OF RULES IN RULE-BASED MODELS OF HUMAN PROBLEM SOLVING

C. Michael Lewis

John M. Hammer

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

To appear in IEEE Transactions on Systems, Man, and Cybernetics.

INTRODUCTION

Many researchers have used rule-based systems to model human problem solving [1,3,6,7,11,12]. Typically, the rule-based system has a large number of rules, each of which has several free variables that were adjusted during the modeling process. For the most part, significance testing of these rules has not been much of a consideration. It should be. It is certainly possible to describe N data perfectly with N rules using a trivial model that simply reproduces the data. While there is no evidence that this has happened in any of the research reported to date, there is a certain danger of overfitting a rule-based model.

In this article we present three methods of testing the statistical significance of rules and other components of rule-based models. Throughout this article we shall assume that the percentage of behavior matched (e.g., commands) is the performance measure of interest. Two of the testing approaches, however, are not limited to this measure. They may be used to study any performance measure, though it may well be possible for a rule to produce a statistically significant effect on one performance measure but not another. The remainder of this article contains a section on notation, three sections on testing by analysis of variance, randomization, and contingency tables, respectively, and two concluding sections on applicability of the various tests and validity of these models.

NOTATION

A rule-based system consists of three components. The first is a set of rules of the form IF condition THEN action. The meaning of the rule is that if condition is true, then action could be taken. For example, the following rules describe behavior at traffic light controlled intersection:

IF	In intersection	THEN proceed
IF	Yellow and arrival at intersection before the light turns red	THEN proceed
IF	Yellow and arrival at intersection after light turns red	THEN stop
IF	Green	THEN proceed
IF	Red	THEN stop
IF	Red and right turn	THEN proceed

Figure 1. Rules for traffic lights.

If the above model can successfully match human behavior, then the rules form a model of the human. Often, the rules are interpreted as a model of the human's knowledge. Intuitively, the better the model matches human behavior, the better the model, all other things equal.

The rules can be transformed easily into a computer program as follows. First, control statements are added that cause the program to examine the rules repeatedly and execute those whose conditions are true. Second, in order to compare model and subject actions, an input statement is added before the first rule. This statement reads the state vector (e.g., the lights, the traffic, short term memory) that was available to the human when his or her decision was made. The program looks something like this:

```
WHILE TRUE DO BEGIN
   READ(STATE);
   IF       (in intersection)                         THEN proceed
   ELSEIF   (Yellow) AND (predict arrival at
              intersection before light turns red)    THEN proceed
   ELSEIF   (Yellow) AND (predict arrival at
              intersection after light turns red)     THEN stop
   ELSEIF   (Green)                                   THEN proceed
   ELSEIF   (Red) AND (right turn)                    THEN proceed
   ELSEIF   (Red)                                     THEN stop
END;
```

Figure 2. Rules in a program.

The second component of a rule-based system is a conflict resolution strategy. It selects the rule to execute when multiple conditions are true. In the above example, a rank-order resolution strategy was shown. It simply uses the first rule that matches. The ranking of rules can then be interpreted as a subject's strategy. Some other conflict resolution strategies include random selection, meta-knowledge, and backtracking. A random selection strategy simply picks at random one of the many matching rules. A meta-knowledge strategy has a higher-level rule-based system that chooses which rule to execute. A backtracking strategy will, if necessary, try all possible matches. It should also be noted that it may be possible to write the rule conditions so that there is always exactly one rule that matches.

The third component of a rule-based system is the input and internal variables. The input variables correspond to external data. The internal variables correspond to human short-term memory, which may be changed by the action part of rules. Both internal and input variables are examined by the condition part of rules.

Evaluation of Models

When comparing subject and model performance, the model is usually run open-loop without any knowledge on subject actions. In other words, the model can simply be treated as another subject. When comparing subject and model behavior, the model is usually run closed-loop as follows. The model has as input the same state vector the subject saw. The model chooses an action, and then it is recorded whether the subject and model agree. Then, the subject's action is used to control the system, and the process repeats. The reason for always following the subject's action is as follows. If the subject and model action differed and both were used, then the state vectors would be unequal after applying these actions. The model and the subject would then be working on different problems, and a comparison of their actions would make little sense.

The following sections on testing rule-based models will specify ways in which the model will be modified and then run. The typical modifications are to delete or modify one or more rules. Running a model, perhaps in a modified form, means to compare its overt behavior, (e.g., commands) to a subject's and determine the percentage in agreement.

ANALYSIS OF VARIANCE

The analysis of variance approach is the simplest of the three approaches for testing rule significance. To use it, each rule in the model is equated with an independent variable. The meaning of the variable is that at its high level, the rule is in the model, and at its low level, the rule is deleted from the model. The rule-based model is then run 2^N times (for each subject), which corresponds to a run with each possible subset of rules present. It must make sense for the model to do nothing, or else the model must be augmented before testing with a special, nondeletable rule that applies when no other rule applies. The resulting data can then be analyzed as an N-way factorial.

To economize on model runs, fractional factorial designs should be used. The full factorial design, proposed above, will estimate the effects of many high order interactions that cannot occur. In fact, the interpretation of an interaction is that the corresponding rules interact. An example would be two rules, the first of which stores some value in a temporary variable and the second of which uses the temporary variable. Such rule interaction is common, but rarely do many rules interact. An inspection of the rule-based model will reveal what interactions could occur. It should be possible to create experimental designs which test only the desired interactions.

The testing of condition components of rules is also possible. In this case the reduction in error attributable to the greater specificity provided by the additional condition can be evaluated. Suppose, for example, that a significance test of each of the conjunctive conditions of a rule is desired. For example,

IF condition_1 AND condition_2 AND condition_3 THEN

Proceeding as before, three independent variables might be equated, one with each of the three conditions. A three-way ANOVA could be run to test each of the three clauses. It would most likely be necessary to estimate the value of the response at the point where all three conditions have been deleted from the rule. Obviously, this process could be extended to cover all of the conditions for all of the rules in the model.

The testing of groups of rules as a whole is also possible. To do this, an independent variable is equated with several rules, not just one as was done initially. The experimental interpretation is that the entire set of rules is either present or absent from the model during an experimental run. This pooling of rules corresponds to a supersaturated experimental design, and may be the only economical means of testing models with many rules. One logical choice for pooled rules would be interacting rules. Another choice would be the modeler's organization of rules into groups (e.g., S-rules and T-rules [6]).

Analysis of variance makes several assumptions, one of which is that error residuals are normally distributed. Moderate departures from this assumption do not produce large deviations in calculated and actual significance levels. If the normality assumption is known or seriously thought to be incorrect, an approximate technique [4] may be used. Simply, the data are replaced with their ranks, and the remainder of the analysis of variance calculations remain unchanged. The significance levels produced by this method are reported to be nearly equal on normally distributed data to that produced by the standard F-test. The rank transformation is more robust with respect to the distribution of the data, though it is not a distribution-free test. Finally, the

hypothesis being tested here is whether the presence of a rule (or some other similar entity) explains a significant amount of variance in the subjects' performance. This significance is independent of the significance of other rules (or other entities) but may be dependent on the conflict resolution strategy. It is important to note the hypothesis because the next section tests somewhat different ones.

RANDOMIZATION

The second approach to testing a rule involves forming a randomization distribution by randomly permuting a rule. Suppose a particular rule is under test. Its action can be replaced by a random action (e.g., a random number generator that chooses commands according to a priori frequencies). The model, with a single modified rule, can be run many times. Its matching performances can be considered a randomization distribution. The model in its unaltered form can then be run, and its resulting performance be referred to the randomization distribution. If its matching were higher than 95% of the randomly generated values, the null hypothesis could be rejected at the 5% level (one-sided). The null hypothesis would be that a random action would be as suitable as the proposed action in the rule under test. The empirically determined significance level is partial in that it is potentially dependent on all the other rules being present in the model as well as conflict resolution strategy.

The condition part of a rule can be tested by a very similar method. There is a minor difficulty in that a random number generator in the condition part of a rule does not appear to make sense. A solution would seem to be to create various mutant conditions by randomly selecting condition clauses from other rules in the model. The null hypothesis being tested here is that random conditions are as suitable as the proposed condition in the rule under test. The significance level attained is partial just as the one obtained in testing actions.

An entire rank order conflict resolution strategy may also be tested by randomization. Basically, a randomization distribution of performances can be obtained by running all possible rank orderings (or a

Monte Carlo sample) of rules. The performance of the model with the original rank ordering can be referred to this distribution as above. The significance level obtained is dependent on the rules.

CONTINGENCY TABLES

Contingency tables are used to analyze nominal data. If the following is a rule-based model:

$$\begin{array}{l} \text{IF condition}_1 \text{ THEN action}_1 \\ \text{IF condition}_2 \text{ THEN action}_2 \\ \qquad\qquad \vdots \\ \text{IF condition}_n \text{ THEN action}_n \end{array}$$

then, a contingency table may be set up as follows:

	action_1	action_2	. . . .	action_n
condition_1			. . .	
condition_2				
$\vdots$			$\vdots$	
condition_n				
NOT (condition_1 OR...OR condition_n)				

Figure 3. A contingency table for rules.

The last row in the table covers the conditions that are not covered by any rules. The observed data fill the table in the obvious way: for a given state vector and subject action, the unique condition which holds is determined, and the cell under the subject's action is incremented. A model that matched the data perfectly would have all zero entries off the diagonal.

Certain restrictions must be met to employ contingency tables:

1. Conditions must be mutually exclusive (2 rules cannot fire at the same time)
2. Actions must be overt
3. Each action must be unique (2 rules cannot issue the same action)

These restrictions may be met in a variety of ways. Mutual exclusivity will be satisfied by any model containing conflict resolution, rank-ordering, or disjoint rule provisions. The unique action requirement may be accommodated by phrasing composite rules in which constituent rules prescribing the same action are joined by disjunction.

The performance of the rules in matching the data can be evaluated with a chi-square or similar tests. The hypothesis is tested whether conditions and actions are independent, i.e., whether there is a significant difference between the proportions given the rules and the overall proportions. As a consequence, rules containing infrequently used actions receive more latitude using these tests than they do under a simple percentage of commands matched measure.

Testing a set of rules is also possible as follows. The null hypothesis is that there is no relationship between the action and the conditions aside from the relationship that is already described by the existing rules. Consider the test for the rule:

IF (x_1 = 1 or x_1 = 2) and (x_2 = 1) THEN action_1

		Action_1	. . . . .	Action_n
X1 = 1	X2 = 1	Delete		
	X2 = 2			
X1 = 2	X2 = 1	Delete		
	X2 = 2			

Figure 4. Table for testing a set of rules.

Two statistics are computed. The first is a maximum likelihood estimate of chi-square, (G^2) for the complete table. The second is a test of quasi-independence [2] for a reduced table in which cells corresponding to rule(s) under test are excluded. This corresponds in a table such as figure 4 to one cell per row for conditions covered by the rule(s). If the original G^2 is significant and the quasi-independent one is not, this implies that the rules capture the dependency of the actions on the conditions. While attractive in directly referencing observables, this method requires large samples with replications of observed combinations of variables. (Unobserved combinations are treated as structural zeros.)

Other Statistics

A nonparametric analogue to the coefficient of determination R^2 is τ_b [8] which may be used to determine the percentage of variance explained in the actions by a rule or rule set.

$$\tau_b = \frac{\sum_i \frac{1}{X_{i+}} \sum_j X_{ij}^2 - \frac{1}{N} \sum_i X_{+j}^2}{N - \frac{1}{N} \sum X_{+j}^2}$$

X_{ij} = table entry in row_i, column j

$X_{i+} = \sum_j X_{ij}$

$X_{+j} = \sum_i X_{ij}$

N = total number of observations

Individual rules, the disjunction of rules issuing a particular action, or the complete rule set consolidated into disjunctions by action can be evaluated in this way. If uncovered observations are excluded, τ_b may be interpreted as the extent to which actions covered by the rule are explained. If all observations are present, a N+1st category should be formed following the distribution of the uncovered actions. This τ_b is interpretable as the extent to which rules explain all the actions.

Values of τ_b are asymptotically related to χ^2 allowing significance testing.

$$\chi^2_{(I-1)(J-1)} = (N-1)(I-1)\tau_b$$

This statistic tests the hypothesis that $\tau_b = 0$, corresponding to the premise that there is no relation between conditions and the actions prescribed by the rule(s).

A similar statistic, PRE (proportional reduction in error) [2] measures the reduction in error achieved by predicting actions based on the rules rather than assigning the modal action under all rules.

$$PRE = \frac{\sum_{1}^{I} P_{im} - P_{+m}}{1 - P_{+m}}$$

where

$$P_{im} = \max_j (P_{ij})$$
$$P_{+m} = \max_j (P_{+j})$$
$$P_{ij} = N_{ij}/N$$

As demonstrated by this potpourri of procedures, a unified technique for testing rule significance based on multinomial sampling is yet to be developed. PRE answers the pragmatic question of gains in prediction. The quasi-independence procedure provides its complement by testing for unmodeled consistencies. Rules can be simultaneously tested in a contingency table but their contributions to rule set performance will remain unknown. τ_b allows both significance testing and estimation of effects but cannot evaluate rule set performance without pooling rules by action.

APPLICABILITY OF VARIOUS TESTING METHODS

For testing the degree to which a model's behavior matches a subject's, all three methods will work. A contingency table is clearly the best, however, since it requires the minimum in computation. Randomization is clearly the worst technique because of the large amount of computation and the partial significance levels it produces. A fractional factorial ANOVA is clearly superior to randomization on both of these points. ANOVA and randomization can both be used to test rules that modify internal, unobservable states. Contingency tables cannot.

For testing overall performance measures, (e.g., time to solution, total errors) only randomization and ANOVA are suitable, with ANOVA preferred. Ordinarily, much more emphasis is placed on behavior than on performance, since behavior is much more difficult to model. There are situations in which testing hypotheses about both performance and behavior is desirable. One might want to show that a certain set of rules will affect behavior but not performance. For example, Morris and Rouse [10] have observed that theoretical training given process control operators often fails to change their performance. It would be interesting to test this concept analytically in a rule-based system. For example, a group of rules might be identified as the intended consequences of theoretical training. The model might be run with and without these rules, using ANOVA to evaluate performance measures and contingency tables to evaluate behavioral differences.

The randomization method can be used on two hypotheses. The first, and more important, is to test the significance of a rank ordering of rules. This would seem to be the only way to test this type of resolution strategy. The second use is to test the hypothesis that part of a rule performs no better than random. This test would seem to be of little use, since ANOVA can test nearly the same hypothesis.

VALIDITY

The previous methods are generally devoted to evaluation of rule performance and do not address the issue of rule validity. Just as a high R^2 does not imply that all terms of its regression equation are significant, a high τ_b does not vouchsafe for the future predictiveness of its rules. This distinction becomes important in the identification phase of rule-based modeling. Unlike identification based on parameter estimation, the identification of rules requires a search of the space of possible rules. An inductive pattern matcher must consider a large number of potential rules. In evaluating identification it becomes necessary to account for the probability of finding rules of comparable quality by chance. To answer this question the structure of the event space (observed combinations of condition variables), distribution of actions, and extent of search (set of possible rules) must be considered simultaneously.

Eilbert and Christensen [5] refer to this problem as contrivedness, "...the tendency of a search procedure to uncover apparent patterns where none exist." They suggest a randomization test for measuring the extent to which a search procedure uncovers contrived rules. The data consist of many pairs of state vectors with subject responses. The state vectors are left undisturbed, but the responses are randomly permuted. The resultant permuted data has reasonable state vectors paired with random responses. Contrivedness is the degree to which the search procedure can make sense of this random data. When many permuted data sets are searched, the search procedure results form a randomization distribution against which the results from the original, unpermuted data can be referred. While the previously mentioned randomization test will give an idea of how opportunistic the search procedure is, it does not say how to refine the search procedure so as to avoid contriving rules.

CONCLUSIONS

This article has identified several ways of testing a rule-based model of human-problem solving. The amount of testing seems to be on a par with the size of the model. Left unresolved for the most part was the problem of contrivedness of automatic rule identification. It seems fitting to close with the description of an interesting and difficult question in identification of rules. As stated earlier, many cognitive models have been built using rule-based models. Sometimes these models are built when the investigator has access to the subject's thinking. This is always the case in developing a rule-based expert system. Other investigators, particularily those running experiments with humans, may have only the data (i.e., commands) to examine.

An important theoretical question is the limits to identification of rules from data that contain response errors. While there has been work in machine learning, it does not seem that anyone has examined this question [9]. It does seem important, because it bears on our ability to construct models. This problem also seems to be very difficult to solve formally. Hence, a preliminary investigation could be done via simulation, as shown in Figure 5. Basically, the approach is to generate some rules and some random stimuli, apply the rules, add noise, and try to identify the rules from the noisy data.

The following would seem to affect identification:

1. the amount of data and its coverage of the stimuli domain
2. size and number of rules
3. the number of times a rule fires
4. the level of noise

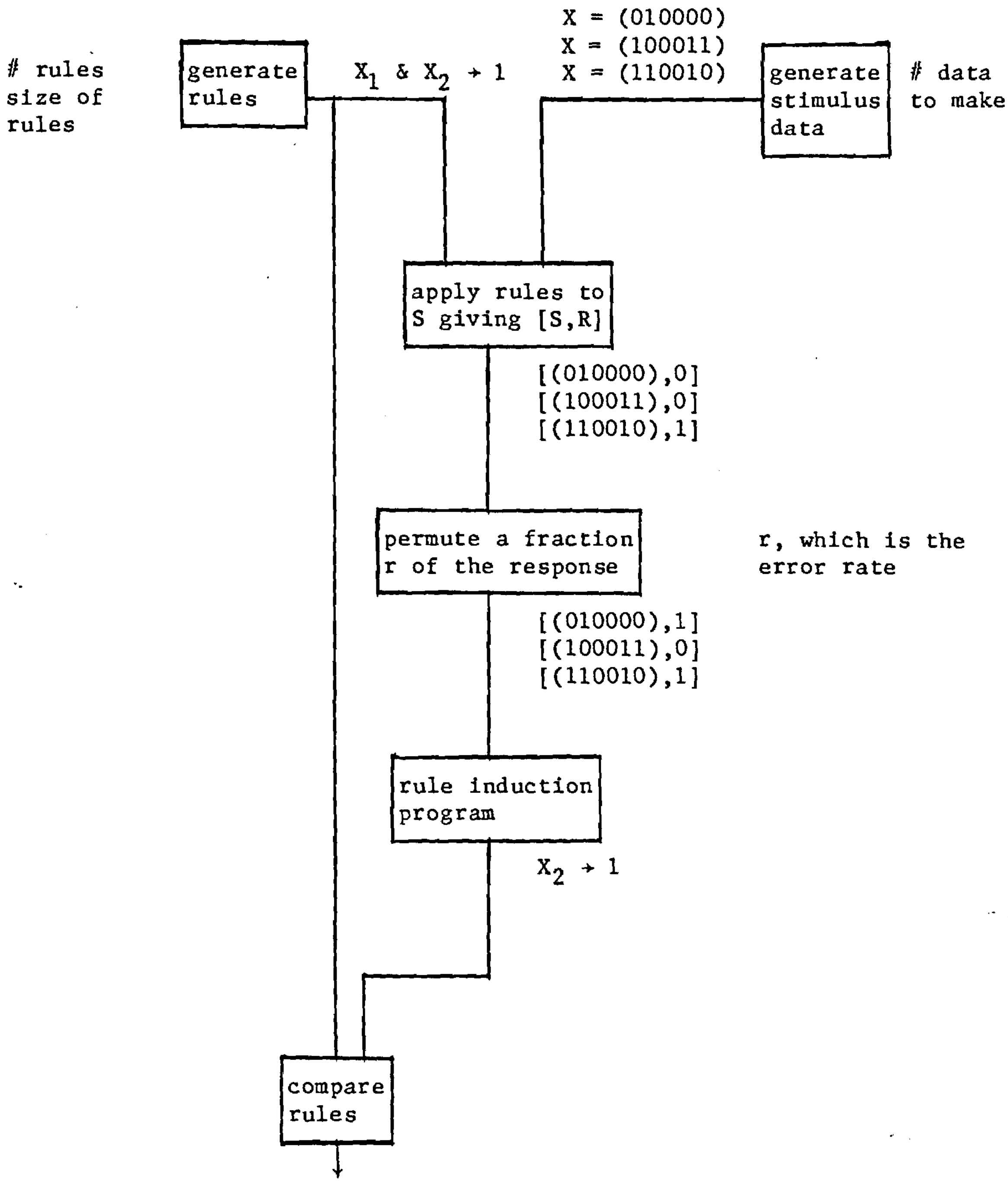


Figure 5. Block diagram for rule induction.
An example output from each block is shown.

It might also be interesting to investigate the addition of oracle variables in rule identification. An oracle variable is an extra variable (beyond the original stimulus vector) that provides information that ordinarily is not available. The first oracle variable might be a single bit to tell whether the response was in error. Another set of oracle variables would identify which rule fired. Yet another set of oracle variables could identify the variables that are part of the rule that fired. While these oracle variables may appear to be practically giving the solution to the identification program, they do not. These variables would be treated the same as any of the real stimulus variables. The identification program would have to infer the meaning of these variables in order to make use of them.

While it does appear theoretically interesting to determine how much oracle variables can add, there are important practical benefits as well. Oracle bits could approximate the hunches of a human investigator. For example, the investigator may suspect certain data to be in error, a certain rule to have fired, or that only certain variables could be influencing the operator's decision. These hunches are a second order human-machine system: the investigator's attempt to identify (with a program) the rules of the human in the first-order human-machine system.

ACKNOWLEDGEMENT

This research was supported by NASA-Ames Research Grant NA602-123. Dr. Everett Palmer is the technical monitor.

REFERENCES

[1] Anderson, J.R., The architecture of cognition, Cambridge, MA: Harvard, 1983.

[2] Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W., Discrete Multivariate Analysis: Theory and Practice, Cambridge, MA: MIT, 1975.

[3] Card, S.K., Moran, T.P., and Newell, A., "Computer text editing: An information processing analysis of a routine cognitive skill," Cognitive Psychology, Vol. 12, 1980.

[4] Conover, W.J., Practical Nonparametric Statistics, 2nd ed., New York: Wiley, 1980.

[5] Eilbert, R.F. and Christensen, R.A., "Contrivedness: The boundary between pattern recognition and numerology," Pattern Recognition, Vol. 15, No. 3, 1982.

[6] Hunt, R.M. and Rouse, W.B., "A fuzzy rule-based model of human problem solving," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 14, No. 1, 1984.

[7] Knaeuper, A. and Rouse, W.B., "A rule-based model of human problem solving behavior in dynamic environments," IEEE Transactions on Systems, Man, and Cybernetics, in press.

[8] Light, R.J. and Margolin, B.H., "An analysis of variance for categorical data," Journal of the American Statistical Association, Vol. 66, 1971.

[9] Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Machine Learning, Palo Alto, CA: Tioga Publ., 1983.

[10] Morris, N.M. and Rouse, W.B., "The effects of type of knowledge upon human problem solving in a process control task," IEEE Transactions on Systems, Man, and Cybernetics, in press.

[11] Newell, A. and Simon, H.A., Human Problem Solving, Englewood Cliffs, NJ: Prentice-Hall, 1972.

[12] Rouse, W.B., Rouse, S.H., and Pellegrino, S.J., "A rule-based model of human problem solving performance in fault diagnosis tasks," IEEE Transactions on System, Man, and Cybernetics, Vol. 10, No. 7, 1980.

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semi-Annual Progress Report

August 1985 - February 1986

William B. Rouse, Principal Investigator

John M. Hammer, Co-Principal Investigator

C. Michael Lewis

Wan C. Yoon

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, GA 30332

*The NASA Technical Officer for this grant is
Dr. Everett Palmer
NASA Ames Research Center MS 239-2
Moffett Field, CA 94035

Introduction

This report covers progress during the six month period from August 1985 to February 1986. During this time there has been considerable progress in the following areas. Identification of the rules of PLANT operators is nearly completed. The aid for the Diagnosis of Infrequent Malfunctions (ADIM) has been programmed in a prototype form. A conceptual basis for preventing aircraft constraint violations (e.g., collisions) has been established in the constraint-based pilot's associate. Finally, the mechanical and electrical work on the aircraft simulator has been completed. The goals of the research are to build aids to help human operators control complex systems. A common approach throughout this research is the use of models to understand and aid operators. For example, the novel fault diagnosis research uses a model of a physical system to aid the human. The constraint-based pilot's associate uses an aircraft model to detect violations in constraints.

The motivation for using system models in aids is to build smart interfaces. One way to produce an intelligent interface is to give the aid an understanding of the physical system. (A second way, perhaps orthogonal to the first, is to give the aid an understanding of the user, perhaps in terms of task or typical use knowledge.) It is possible to produce the illusion of intelligence by using an expert system based on shallow reasoning. This is not the approach taken in this research. Instead, aids based on models would seem to produce interfaces that are truly intelligent and more reliable.

Inductive Rule Identification of Human Operators in Plant

The research problem is to identify the rules used by human operators to control PLANT, a generic process control simulation. The data consists of a chronological sequence of state vectors and the operator's response to these states. The state vector consists of all displayed information about PLANT.

It is believed that operators used goals when controlling PLANT. Since these goals are unobservable and usually time-varying, a discrete control model was used to infer operators' goals. The inferred goal then augmented the state vector. It was necessary to augment the state vector in this way, because the inference program INDUCE only works with information provided to it (it cannot infer missing variables).

The current status of this project is that INDUCE has finished analyzing all of the data. There remains some post-processing that must be done. First, the rules must be merged, since INDUCE was able to analyze part of the data at one time. Second, the data coverage and selectivity of rules must be determined to retain only the best rules. Finally, the rules are organized into a tree and the final selection is made by hand.

Aid for the Diagnosis of Infrequent Malfunctions (ADIM)

The research problem is to build a fault diagnosis aid that understands the system and human fault diagnosis well enough to aid the human operator who is diagnosing a novel failure. The aid contains a qualitative model of the Orbital Refueling System (ORS), a NASA-designed shuttle payload for refueling orbiting satellites. By using the qualitative model, the aid (or the human) is able to deduce component states from given data (pressure, temperature, flow, valve commands, and other known component states).

The aid is also designed to compensate for various suboptimalities that occur when humans troubleshoot. A widely recognized suboptimality is an incorrect understanding of component and system operation (incorrect mental model), which the qualitative model should remedy. Other important suboptimalities include incomplete hypotheses set generation and "cognitive tunnel vision" (fixation on a single hypothesis without considering any others). To remedy these, the aid will generate and display its own hypothesis list using the qualitative model. The aid will also attempt to infer the operator's hypothesis from the commands issued. The aid will then apply plausibility tests to the operator's inferred hypothesis.

The status of this project is that the qualitative model is finished. Work is now underway to program the user interface. We expect to send to NASA-Ames a demonstration program (in Franz Lisp) in May.

A Constraint-Based Pilot's Associate

A "pilot's associate" may be based on either task analysis or physical constraints. The first and relatively common approach uses mission models to recognize the pilot's task from a rich and complex set of uses. Natural language understanding techniques are useful for recognizing situation and intent. After recognizing the pilot's intent and the situation, the aid can use this information to plan, provide advice, select displays, take action on its own, etc.

The task model approach uses pilots as a source of expertise. One possible serious problem is that pilots may disagree with each other about what is proper. This disagreement may be so significant that each pilot might require a personalized associate. Identifying personalized associates might be magnitudes more complicated than building a generic associate.

A second approach to the pilot associate is to view flying as operation within constraints. Further, operation of the aircraft anywhere in the feasible region (thus allowing for pilot judgment) could be viewed as satisfactory. Only constraint violation is a concern for the associate. Most constraints are based on physics, or relatively crisply defined laws and policies. It may be that there is only a single set of constraints for a given model of aircraft, in contrast to multiple pilot preferences.

The constraint approach uses designers as a source of expertise. To within a safety margin, most designers would seem to agree on constraints. Furthermore, designers are available during the aircraft design, and it would seem wise to use their knowledge in the associate.

An appendix ("Preventing maneuvering errors") contains a description of a constraint-based associate. This working note outlines the philosophy of such an aid. While this philosophy is still adhered to, the notion of using algebraic manipulation has been abandoned. The only models suitable for use are numerical. While the technical feasibility of the approach has yet to be demonstrated, it may be that this approach would eliminate aircraft crashes and solve in a general way the problem of when automation should take over from the pilot.

The original plan for demonstrating a constraint-based associate was to study aircraft landing. As it turns out, this choice was unfortunate. Considerable time was spent studying aerodynamics and stability/control to build an aircraft model. It would appear that these models are relatively well understood. Consequently, it would appear relatively straightforward for an airframe manufacture to produce this kind of aid. Consequently, a new problem based on a more symbolic task has been chosen.

DC-8 Flight Simulator

Mechanical and electrical changes to the DC-8 Flight Simulator are complete. The simulator has been reassembled, and all sensors, indicators, and displays are electrically connected to terminal boards at the base. Much of this work was completed by Georgia Tech Research Institute personnel, who had the necessary fabrication facilities. They were also expensive. It is fair to say that we did not fully understand the knowledge, time, or finances required to build a simulator. Nevertheless, most of the investment has now been made and the simulator (see photo below) is close to being ready to support research studies.

Two phases remain to be done. The first is to interface the simulator inputs to the A/D converter and thus to the VAX. The second is to use the VAX simulator outputs to drive the displays.

PREVENTING MANEUVERING ERRORS

OR

WHY $F=ma$ and $C_l < C_{l_{max}}$

IS BETTER THAN RULES LIKE

```
IF   (RADAR_ALTITUDE < 1500)
AND  (150 < AIRSPEED < 165)
AND  (FLAPS = 0)
AND  (|ROLL| > 10)
THEN ALERT_STALL_WARNING;
```

John M. Hammer
Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

November 27, 1985

ABSTRACT

A new approach to error-tolerant system design is proposed. Its source of expertise is in symbolic interpretation of physics equations, not in rules.

INTRODUCTION

The problem is to build an aircraft which tolerates human error. The approach is to build a monitoring system that observes the aircraft and the changes made to it by the pilot. Conceptually, the aircraft must operate within an envelope. The monitoring system has a model of the aircraft, the envelope, and pilot plans. It uses these to predict possible envelope violations.

This automation approach is termed human error tolerance. The observation behind it is first, that human error cannot be eliminated. The introduction of automation displaces but does not eliminate human error. The automated device must still be monitored by the human. Humans are poor monitors and sometimes perform poorly when taking over after automation fails.

Rather than use automation in a futile attempt to eliminate human error, we propose to use it to build an error-tolerant system. Under this arrangement, the human is allowed to operate the system normally. The automation monitors the human for errors according to a computer-based representation of what should happen. The monitor asserts itself only in the event of an error.

EXPERT SYSTEMS: THE ROAD NOT TAKEN

An approach often espoused for this kind of problem is to build an expert system for monitoring. Such a system is usually contains a set of shallow rules that examine the current aircraft state for possible errors. This seems to be the approach taken by the aerospace community.

There are disadvantages to such an approach. First, the future must be predicted if the action is to be taken by the pilot in time. Pilots usually agree that their planning must remain "ahead of the aircraft." Thus, the values needed by the expert system must be future, predicted values. Second, how can a large rule-based system be verified to work? Program testing, which is the usual method of verifying software, demonstrates only the presence of bugs. Further, the rules are not well organized for the purposes of proof. Knowledge has been scattered about, which often causes unusual side effects in unusual situations. Unanticipated modes of operation and expert system brittleness combine to cast a shadow of doubt on the value of this aid.

My subjective experience with such an expert system is the basis for these statements. While the system works, there was always some uncertainty that the program would behave correctly on new data. Often this uncertainty would be on the correct value for a parameter in a rule. For example, should the clause be (altitude < 1500) or (altitude < 2000)? Another frequent uncertainty was whether the correct variable was being used or if a condition might be missing.

While an expert system should work some large fraction of the time, it is not clear that its reliability will approach that of traditional aeronautical systems. Thus, the question arises as to whether there should be an investment in an approach that might have practical reliability limits.

A SYMBOLIC, AERODYNAMIC INTERPRETATION OF MOTION

The proposed alternative to a shallow (i.e., surface cues only) expert system is to use the aerodynamic equations of motion to interpret the aircraft maneuvers. Simply stated, the expertise to be used is the set of equations of motion. The system block diagram appears in Figure 1.

The basic question which the symbolic interpreter (MACSYMA, REDUCE) is repeatedly asked is the following:

(1) For a given control (e.g., flap position)

(2) For a particular constraint (e.g., $C_l < C_{l_{max}}$)

(3) Is there a point in time in the near future (say 90 seconds) at which flap position could cause the constraint above to be violated given the following: the current aircraft state (from sensors), other predicted control actions (from script), and current model of motion.

(4) If the symbolic manipulator indicates a problem, it will give a specific time or time range when the constraint will be violated.

This basic question is asked of all control-constraint pairs.

The purpose of the scripts is to fill in default values for pilot actions. In essence, the scripts allow fixed values for all pilot control actions except for the one being solved for. There is an assumption here that the one variable-at-a-time approach will work.

Some examples of constraints include fuel quantity, lift, airspeed, g-forces, and a complex ensemble of conditions for landing.

ADVANTAGES OF THIS APPROACH

This approach has several advantages over an simple expert system. The first is that the system is more explicit. The predictions of the future are explicitly done. The assumptions of future pilot actions are explicit in the script. The aerodynamics are explicit. All this explicitness should make the aid verifiable and perhaps smaller and separable.

The second advantage should be that the knowledge is more general and robust.

The final advantage is that this approach solves the difficult problem of when automation should step in and take over. Note that the output of the symbolic problem solver is, in effect, a plan. It is the last possible time that a control action could be taken to avoid a constraint violation. As the aircraft approaches the constraint, the pilot can be warned. As the aircraft starts through the constraint, the monitor can step in and take over using this plan.

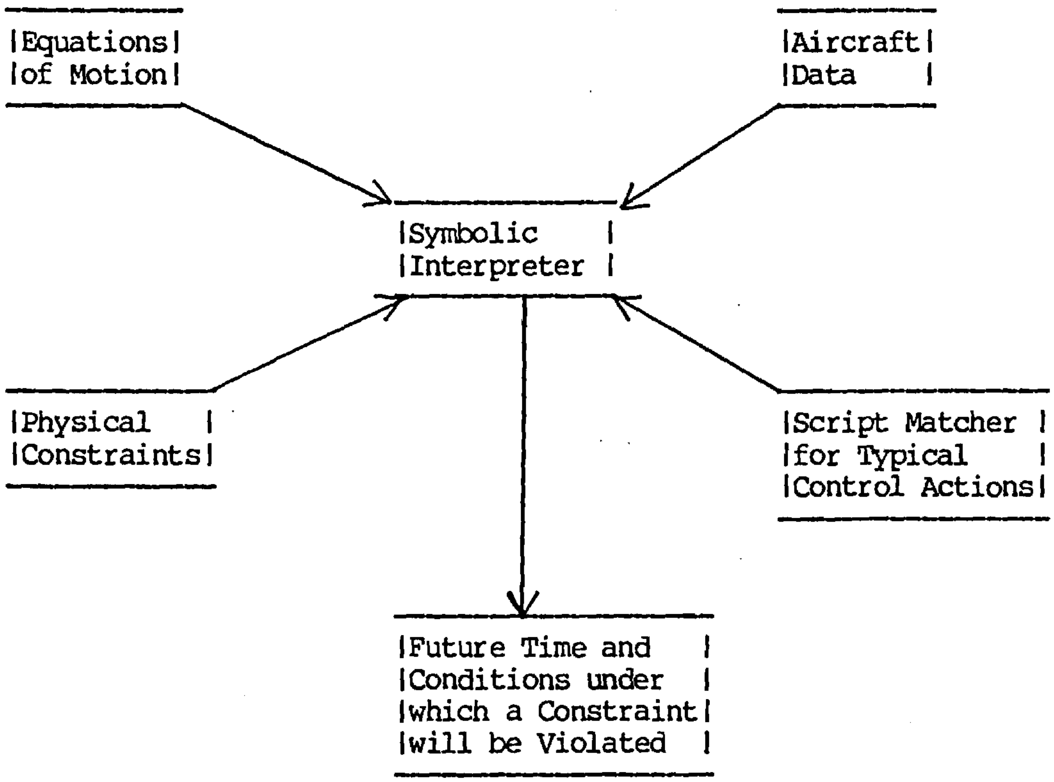


Figure 1. Block diagram of equation-based interpretation.

E-24-637

NASA Grant NAG 2-123*

PILOT INTERACTION

WITH

AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Reports

March 1986 - August 1986

September 1986 - February 1987

John M. Hammer, Principal Investigator

Wan C. Yoon

Vijay Vasandani

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

* The NASA Technical Officer for this grant is Everett Palmer, NASA-Ames Research Center, Moffett Field, CA 94035.

INTRODUCTION

This report covers progress during two periods: March 1986 - August 1986 and September 1986 - February 1987. During this time substantial progress has been made in two areas. The first is Wan Yoon's Ph.D. thesis, "Aiding the Operator during Novel Fault Diagnosis." The second is a newer initiative, "A Model-based and Constraint-based Warning System."

The following were published during this period:

Journal articles

Yoon, W.C. and Hammer, J.M., "Aiding the operator during novel fault diagnosis," to appear in IEEE Transactions on Systems, Man and Cybernetics, 1987 (Appendix A).

Yoon, W.C. and Hammer, J.M., "A deep reasoning aid for aiding deep reasoning fault diagnosis," to appear in Human-Computer Interaction II, (G. Salvendy, ed.), Elsevier: Amsterdam (Appendix F).

Technical reports

Lewis, C.M., Identification of Rule-Based Models. Technical Report 86-5, Center for Man-Machine Systems Research, Georgia Institute of Technology, Atlanta, Georgia.

Conference papers

Yoon, W.C. and Hammer, J.M., "Aiding the operator during novel fault diagnosis," Proceedings of the IEEE 1986

International Conference on Systems, Man and Cybernetics, Atlanta, Georgia, 1986.

Technical Effort

During the first six months of this period, only Wan Yoon was supported. During the last six months, all three personnel were supported. During the summer of 1986, Dr. Hammer worked on the DARPA/AF Pilot's Associate program. Many of the interface concepts in the PA program were developed under NASA-Ames sponsorship. It was clear by the end of the summer that direct competition with this program was not possible. The PA program has more funding. The PA program can implement any aiding process that depends on knowledge acquisition from pilots. It cannot stop to answer basic research questions, although many have arisen during implementation. These unresolved questions are excellent topics for this grant because they are both relevant and realistic.

Relation to Earlier Work

The current research is focused on detection of human error and protection from its consequences. The first work in this area under this grant was [Hammer, J.M. "An intelligent flight-management aid for procedure execution," IEEE SMC 14(6), 1984], which described a program for monitoring pilot errors by comparing pilot actions to a script. There were two dimensions to this work. First, it dealt primarily with routine errors (slips) that occurred during checklist activity. Second, the model to which operator actions were compared was a script. There was no model of the aircraft or any part thereof.

Current research is an extension along these two dimensions. The ORS novel fault detection aid uses a sophisticated device model rather than a script. Since this aid has been used to study novel fault diagnosis, the errors committed are bad decisions, not slips. Although error detection is not currently implemented, the plans for it are discussed in [Yoon and Hammer, 1987] and later in this report.

The newer initiative, the model-based and constraint-based warning system, uses an even more sophisticated device model and is to prevent all types of error, not just slips or bad decisions.

PROJECT ORGANIZED BY MODELS OF DEVICES AND HUMANS

The principle that organizes this project is that model-based reasoning be the basis for aiding the human operator of an aerospace system. There are two models. First, the aid will contain a model of the device. The aid uses the device model to produce information for the operator. Second, the information produced for the operator is based on a model of human information processing. More specifically, the aid produces information that the operator needs and that is difficult to produce. What is difficult to produce is determined from the human information processing model.

The principle can be seen quite clearly in the novel fault diagnosis research. First, the aid has available to it a qualitative model of the orbital refueling system. Second, the aid uses this model to display information about what the ORS does normally (N aiding), what it is estimated to be doing (O aiding), and the difference between normal and observed behavior (O-N aiding). It can easily be argued that the unaided human operator must use at least some of this information in order to diagnose effectively a novel failure. This means the unaided operator must produce the information internally. It is difficult for the unaided operator to produce this information.

Model-based aiding can also be seen to organize the model-based and constraint-based warning system. The function of this aid is to keep track of the present and future constraints on the system and to detect present and future violations.

Central to this warning aid is a model of the physical system. Constraints arise from both physical and operational considerations. The model of the human is used to provide operational constraints and potential future inputs to the device. This model is actually part of the aid. The model also tells us that the operator does not or cannot consider all of the constraints when choosing an action. This model is not part of the aid. It is the reason that aiding was implemented.

Motivation for a Model-based Approach

The motivation for a model-based approach is two-fold. First, model-based aiding is aimed at a technological breakout through the use of artificial intelligence in device modeling and, to a lesser extent, in the operator intent inferencing. We believe this approach will yield larger system performance improvement than a more empirical approach.

The second motivation is to use what is known about human information processing and cognitive psychology to do cognitive engineering. Fortunately, exact predictions about human information processing are not always required. If some require processing is known to be difficult or error prone, then aiding (using artificial intelligence) should be investigated.

Artificial intelligence and cognitive psychology (or at least that part that is model-oriented) are close enough to use the same technical language. A consequence of this is a synthesis between the human and device models. Another consequence is an increased emphasis on the artificial intelligence technology of the aid.

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

The technical status of this effort is described in Appendices A and F. The remainder of this section describes the technical progress during the reporting period and future plans.

Technical progress

In February 1986, the ORS simulation was just a simulation connected to a display. There were plans for aiding, but no implementation. Since then, the following have been completed.

1. The code for O, N, O-N, and O-H aiding was written and debugged.

2. A preliminary, observation evaluation of unaided problem solving was conducted. The results are described in Appendix A.

3. Three experiments to evaluate N, O and O-N, and O-H have been planned. The first two have been completed and the results are described in Appendix F.

4. The training materials for the experiment were produced. The materials had to be carefully prepared and refined for two reasons. If they allowed too much practice or were otherwise too successful, the subjects might no longer use knowledge-based reasoning. On the other hand, too little training would not allow the subject to understand or interpret the basics of fluid flow.

Future plans for the ORS simulator

Considerable effort went into the construction of the ORS simulation. Relatively less effort was required to produce the existing aids. We would like to capitalize on this by studying a variety of research questions using the ORS simulator. The following are a list of potential problems to investigate.

1. Add and improve existing aids. We have observations from our more recent experiments that suggest more about aiding the operator. The O-N aid, which points out pressures which

differ between the observed and normal system, is useful primarily at the beginning of diagnosis. This is because it guides the diagnosis to the proper locale but is of less assistance thereafter. (In contrast, the O aid, which shows equal pressure paths and mass flow paths, appears useful throughout diagnosis.)

These observations suggest that operators need an aid during local testing. During local testing we have observed two operator deficiencies. First, the operator may incompletely test a local region (which does contain the fault), and then moves to another part of the ORS for testing. This greatly lengthens the time to diagnose. The operator needs to know when a locale has been completely tested. Second, the operator could probably benefit from seeing a list of suggested hypotheses. While suggesting hypotheses is computationally intractable for the entire ORS, it may be reasonable for small locales. In fact, if the aid were able to eliminate infeasible hypotheses as data were collected, the display of remaining hypotheses may keep the operator from leaving the locale prematurely.

Another observed problem is that the operator can choose good hypotheses but cannot effectively test them. This suggests a hypothesis testing aid which converts a specific hypothesis into a series of actions that test it. Interpretation of the results could optionally be included in the aid as well.

Another aid would prevent fault masking. It is possible to configure a malfunctioning system so that its fault is not apparent. For example, if there is a leaking valve, this failure can be masked by closing another valve in series with it. The operator can mask a fault through a series of changes and then be unable to unmask it. Unmasking requires only undoing all the changes, but the operator may not be able to remember them. It would be relatively simple to make available a command to return to the most recent state where abnormal behavior was observed.

2. Inference of operator intent, especially of hypotheses. For the aid to know the operator's intent would be useful in several advanced aiding methods described below. (Intent inference is not directly useful in and of itself.) The first step in understanding operator intent is to understand the process of hypothesis formation in the diagnosis task. While a preliminary description of this is in the paper in Appendix A, it is too subjective to be implemented on a computer. A more detailed examination of the verbal protocols currently being collected should yield a process description of diagnosis.

Given an objective process description, it would then be possible to detect the occurrence of decision-making biases during fault diagnosis. In fact, virtually all of the effort to do so is front-loaded into the intent inference work. Once an operator hypothesis is known, it would be relatively easy to test it for plausibility or keep track of how long the operator maintained it.

To build a training system for the ORS requires an intent inferencer. It may have to be modified to reflect a student's reasoning process. A more systematic approach, however, would be to let the intent inference be a prescriptive model or descriptive model of an expert. To accommodate the novice, a buggy model of dynamic process understanding could be used. This buggy model would be analogous to the buggy models of subtraction and programming that have already been developed.

3. A failure novel to the aid. Currently, the aid's model has a representation for every possible failure mode of every component. It would be interesting to give operators a failure that the aid's model does not

represent. This truly novel failure would occur after the operators had been aided on a series of more routine problems. It is important to know if the operators could determine when the aid was wrong.

MODEL-BASED AND CONSTRAINT-BASED WARNING SYSTEM (MCBWS)

MCBWS is a warning system for detecting present and future constraint violations in aerospace systems. For demonstration purposes, the fuel system of the F15 was chosen (Appendix E). The warning system contains a model of the fuel system and the physical and operational constraints on it. The purpose of this research is to demonstrate an electronic cocoon to surround the operator. The boundary of the cocoon is determined by present and future constraints. The system will be allowed to operate anywhere within the cocoon. Drawing near the boundary will cause an error message. Once demonstrated, the principles should be applicable to a wide variety of aerospace systems.

Motivation for the Warning System

Flying an aircraft requires thinking about the future. Avoiding error means avoiding constraint violations. Thus, it would seem that avoiding future constraint violations is central to avoiding error. Our view of flight is that it is a problem of remaining within the constraint envelope. The remainder of this section describes the implementation and current status of the project.

System Engineering

The two primary components of the warning system are the fuel system model and the constraint identifier. The fuel system model is capable of answering questions such as whether a particular constraint is currently violated or will be violated either now or in the near future. Predictions about a future constraint violation require both the constraint itself and likely fuel system inputs from now until the future point. Both of these come from the constraint identifier. Most constraints are the result of operator plans. The constraint identifier uses both the aircraft state and operator actions to select pilot plans. Associated with these plans are

- predictions of the future input actions to the fuel system

- constraints that must hold during the plan

- future plans that may occur, along with a description of the situation in which they will occur

As can be seen, once a plan is identified, its actions, constraints, and future plans are known. From future plans, future actions and constraints can be determined. Obviously, this forward chaining process can be continued as long as necessary or feasible.

Current Status

All of the technical effort has been devoted to building the fuel system model, which is more fully described below. The constraint identification code has received no attention because 1) I know how to do it from working on the Pilot's Associate program; 2) it is not hard to construct a plan recognizer for those plans relevant to the fuel system; and 3) the constraint recognizer cannot be tested without the fuel system model.

Fuel system model

The fuel system model is organized as a set of components. Each component is connected to other components or to inputs or outputs at the boundary of the system. Components have one or more behaviors, each of which is described by a set of equations or inequalities (termed algebraic relationships)[1]. These algebraic relationships describe the relationships between component inputs, outputs, and state variables. The relationships are symbolic and could be interpreted either quantitatively or qualitatively. If symbolic processing cannot answer a question about constraint violation, a numerical answer could be determined.

One of the basic operations is to solve the fuel system, which means to determine the behavior of each component. This occurs as follows. Each component has several mutually exclusive behaviors. First, find the subset of behaviors that is feasible. Some behaviors can be shown infeasible immediately because at least one algebraic relationship in the behavior is violated by other algebraic relationships known to be true. For each

1. The constraints that arise from operational or physical considerations will be expressed in a form identical to the algebraic relationships that describe component behavior.

feasible behavior, assume that behavior is valid. Then, recursively attempt to solve the remaining components for their behaviors. This is a simple depth-first search through a space that is constrained by the behaviors of the components.

What has been written is the following. A slot-filler representation has been adopted for component descriptions (Appendix D). A set of routines that solves for the component behaviors has been written. A set of routines for manipulating a quantity space has been written. We were unable to reuse Wan Yoon's quantity space code for the following reason. His code uses property lists to store information. A change to a property, even if done within a function, is globally visible. It is as if properties are stored in global variables. Properties are not automatically undone during a search backup, which makes them undesirable.

Future Plans

The following must be done:

1. Build a model of the fuel system. This requires that we understand the fuel system: the types of pumps, the components that are not shown on the figure, etc. This understanding must then be encoded in the representation language and debugged.

2. Build the constraint identifier. This will require knowledge engineering with pilots to determine operational and physical constraints. These will be attached to plans, which will also require identification and duration conditions.

After this much development, the system can be demonstrated to detect current constraint violations. As described earlier, this is not sufficient to meet the need to prevent the consequences of pilot error. Reasoning about the future is also required. The second part of the project will develop this capability and will parallel the the first part.

1. Extensions for reasoning about the future. The model (the reasoning component, the device representation, and the fuel system description) must be enhanced to allow prediction of the future. The inputs to the model then become: the current system state, the predicted pilot inputs to the system, and the constraint(s) to be tested for potential future violation. It is possible for the model to output either yes or no. A no output means that there is no way that the constraint will be violated. A yes output will mean that there is no way to avoid violating the constraint. The most likely expected output from the model would be another list of constraints. This output list would have to hold for the input constraint to remain unbroken. The output constraints in general will have to hold at times not later than the input constraint. This is because if violating the output constraints would cause the input constraint to be violated, then the output constraints must be violated first. The output constraints hold at times closer to the present than the input constraints.

2. The constraint identifier knowledge representation will need to be enhanced. Each plan will also need to have (1)

potential future plans plus the conditions under which each future plan would occur; and (2) the predicted pilot input over the duration of the plan.

Appendices

A Aiding the Operator During Novel Fault Diagnosis

B Instructions for Wan Yoon's experiment, parts 1 and 2

C Problems worked after part 2 training (both training problems and experimental problems)

D Component knowledge representation

E F-15 fuel system

F A Deep-Reasoning Aid for Deep-Reasoning Fault Diagnosis

A

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

Wan C. Yoon and John M. Hammer

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

ABSTRACT

The design and philosophy are presented for an intelligent aid for a human operator who must diagnose a novel fault in a physical system. A novel failure is defined as one that the operator has not experienced in either real system operation or training. Because the fault is novel, the human must reason using causal knowledge. The aid contains unique features that support such reasoning. One of these is a qualitative, component-level model of the physical system. Both the aid and the human are able to reason causally about the system in a cooperative search for a diagnosis. The aid has direct access to the operator's hypotheses when the qualitative model is used. Because of this, various decision-making suboptimalities and biases can be detected and mitigated by the aid.

INTRODUCTION

In highly automated systems, the human operator is primarily a monitor and supervisor [Rasmussen 1983, 1984]. An important monitoring function is diagnosing equipment faults, a difficult task in automated systems. The current approach to fault diagnosis is to train the operator to deal with relatively common faults. The training might teach the operator to use symptoms to distinguish faults and to follow procedures to correct them. While this approach should be successful with common faults, it does not support diagnosis of novel faults.

A common sense but unsuccessful approach to help operators diagnose novel fault is to teach them the principles of operation of the system. With this theoretical knowledge, the operators should be able, in principle, to diagnose any failure. Unfortunately, there is little evidence that theoretical knowledge helps operators diagnose failures [Morris and Rouse 1985a, 1985b]. A logical consequence of this observation might be to put theoretical knowledge into the aid rather than the operator.

Recently, there has been much interest in supporting the human operator via expert systems for diagnosis. To be sure, this approach will improve the system performance on relatively common failures. As for novel failures, many expert systems for diagnosis [Shortliffe 1976, Miller, Pople, and Myers 1984] are based on shallow reasoning: a set of symptoms suggests a diagnosis. This mapping is not explicitly based on a system model. Consequently, such systems are subject to the same limitations as training and procedures. The designer may have to anticipate the failure for the expert system to solve it correctly.

Aiding from Deep Reasoning

In contrast to the above, our aid is based on deep, causal reasoning about the system. There are several advantages to this approach. First, novel fault diagnosis is normally considered to be knowledge-based reasoning [Rasmussen 1983]. Hence, it seems appropriate for an intelligent aid to reason causally. Second, this approach should be more reliable and robust. The system knowledge is represented at the component level. Because components are small and comprehendable, it should be possible to create representations that are correct, perhaps even provably so. These points support the belief that causal reasoning can cover a wider range of faults [Davis 1984].

In spite of the power of the intelligent aid, we believe there are several reasons to keep the human in command of the problem solving. First, diagnosing a novel failure may require the human to extend the aid's model. Second, when diagnosis involves operating the system (e.g., opening valves, starting motors), it would be better to leave these operations to the human. Third, causal reasoning is slow because the diagnosis problem is a combinatorial search. It may be that the human and the aid may be better able to find a solution cooperatively than either can alone. This is possible, even necessary, for two reasons. The human has better pattern recognition capabilities and can make inductive leaps. Second, the human may need to resolve ambiguities inherent in the aid's model.

Decision-Making Biases

The aid is designed to mitigate human suboptimalities that occur during decision-making and troubleshooting [Wickens 1984]. Two categories of suboptimalities used here are knowledge-limited and cognition-limited. The knowledge-limited suboptimality is simply that the operator does not fully understand the system. Obviously, the aid's model is a basis for compensating for this problem. There are many cognition-limited suboptimalities, which are discussed fully in a later section. The aid is designed, however, to prevent suboptimalities from occurring as well as detect and announce any that do occur. It should be noted that detection of suboptimalities requires a system model. Without a model it is logically impossible for the aid to interpret what the operator is doing. Thus, the system model is fundamental to aiding.

Motivation for this Research

There are several justifications and motivations for the research in this area. The first is to explore a new basis — qualitative models — for aiding humans in a domain for which there are few aids. Specifically, we wish to evaluate the suitability of qualitative models as the internal model of the aid. Many claims [Gentner and Stevens 1983; Rouse and Morris 1986] have been made that humans reason qualitatively about physical systems. The implication, which will be tested, is that qualitative models are useful as models in aids. Second, we wish to form a more detailed understanding of human diagnosis of novel faults. This presumably significant role for humans will be studied initially with observational methods, including verbal protocols.

In the subsequent sections of this article, we will review some relevant research on novel fault diagnosis, discuss the context of our experimental task, and discuss the qualitative model in our aid and its expected effects. In the final section, we will discuss the suboptimalities of interest and the methods to mitigate them.

REVIEW OF NOVEL FAULT DIAGNOSIS IN COMPLEX SYSTEMS

The literature on novel fault diagnosis in complex systems is limited. The section will have three parts. The first is empirical research on the effects of training on diagnosis. The second is Rasmussen's system engineering approach to the information needs of operators. The third is Wohl's performance model for predicting diagnostic times for novel failures. The last is the human information processing view of problem solving, which is similar in some ways to novel fault diagnosis.

Shepherd et al. [1977] have studied the effects of training on the errors operators committed while diagnosing familiar and unfamiliar failures. There were three kinds of training. The first was "no story," which amounted to a brief introduction to the control panel instruments. The second was "theory," in which the operation and flow of materials was explained. The third was "rules," which included the above theory training plus a set of proceduralized rules for diagnosing failures. After this training was administered, the three groups were tested. All three groups were significantly different, with rules best and the no story group worst on accuracy. The groups were then trained by examples to diagnose faults, and a second test revealed no differences between the groups. Later, all groups were tested again with two sets of faults — familiar and unfamiliar. Familiar faults were diagnosed equally well by all groups, but unfamiliar faults were diagnosed best by the rules group.

An experiment on the effects of training on operator control of a simulated process control plant has been conducted by Morris and Rouse [1985a]. One situation examined was the diagnosis of novel failures for which some of the subjects had sufficient theoretical training to diagnose the failure.

The system controlled was a network of fluid tanks. Fluid was pumped from these tanks through valves to neighboring tanks. Two novel failures were studied: a tank rupture that caused a loss in fluid, and a safety system failure that caused the system to shut down when it was not in danger. The experimental results did not show any differences due to training. Nearly all subjects were able to diagnose the tank rupture, and only half were able to diagnose the safety system failure.

System Engineering and Complex Diagnosis

Rasmussen [1983] has discussed operator control of complex systems in terms of three levels of information processing: skills, rules, and knowledge. Skill-based performance applies primarily to automatic, sensory-motor tasks that proceed without conscious control. One characteristic of such performance is that it is not decomposable or verbally expressible (for example, one cannot verbalize the skill of riding a bicycle).

The rule-based level is the second level of processing. A rule is a direct mapping from a set of input symptoms to a diagnosis or action. While performing at this level, the operator does not make recourse to causal models. Rule-based reasoning can be verbalized, which distinguishes it from the previous level.

The knowledge-based level is most relevant to the research reported here. Knowledge-based reasoning must be applied when novel failures occur. Neither skill-based or rule-based behavior should be used, and hopefully, the operator realizes this (but there is no guarantee). The operator's control occurs by first forming a goal and then a plan consisting of actions that lead to the goal. The plan is evaluated and perhaps modified by a combination of mental simulation or actual actions taken on the machine. Mental simulation relies, among other things, on the operator's mental model of the system.

Rasmussen [1985] has discussed functional and causal reasoning in diagnosis and control of complex plants during novel failures. Physical systems may be represented along a hierarchical, causal-functional continuum. The causal end of this dimension is a description of components according to their local behavior and their physical and structural location (much like a qualitative model). The functional end of the dimension is a description of aggregates according to their function or purpose. In highly automated systems, the operator also needs to know the intent of the automation, since it can change both the function and structure by its own action. The implications for novel fault diagnosis are the claims that an operator needs a multilevel display for intention, function, and causation. The motivation for this is that diagnosis begins at a functional level and moves toward a causal level as the diagnosis becomes more precise.

Maintenance Complexity

Wohl [1982] has observed that electronic troubleshooting in complex equipment operates in two modes. This first mode is for routine failures, which account for 65-80% of all failures. These are repaired relatively quickly. The second mode is for novel failures, which require substantially more time to diagnose and lengthen substantially the mean time to repair. A model for predicting the frequency distribution of novel malfunction repairs has been developed and tested. The model has three parameters: an equipment complexity index, which is the average connectivity of a component; second, an average time to test a component; and third, a parameter that describes how diagnostic interpretation becomes geometrically more complex with each diagnostic test. The test of the model showed a correlation of $r=.98$ between measured and predicted mean time to repair for fourteen different electronic systems. In a related article, Wohl [1983] observed that the model predicted an infinite mean time to repair when the equipment complexity index exceeded 7.5. An infinite mean time to repair simply means that some malfunctions are never diagnosed. An equipment complexity index of 7.5 means that the average component is connected to 7.5 other components. This limiting value is close to the chunk capacity of human working memory. This result is consistent with the often observed relationship between connectivity and diagnosis complexity.

Complex Diagnosis and Human Problem Solving

Much of the research on problem solving would appear to be relevant to novel fault diagnosis [Newell and Simon 1972]. We briefly review here the human information processing approach to modeling of problem solving and then discuss how novel fault diagnosis differs from it. The information processing approach is centered around the idea of a *problem space*, which is the human's representation of the key characteristics of a problem. The subject is given an initial and goal state in the problem space and a set of operators that transform the problem from one state to another in the problem space. Usually, the states and operators are crisply defined. Often, there is a metric for the difference between a given state and the goal state. This metric can be used as a heuristic for selecting the operator that moves the greatest distance toward the goal.

The behavior of a human is modeled by a production rule system. Each production rule contains a condition and an action. The condition is a boolean expression on the features of the problem space, some of which are in the human's working memory and some of which are externally perceivable. The potential actions are working memory changes or operators as described above.

Clearly, novel fault diagnosis is a special case of problem solving. The specializations are as follows. First, the human operator must realize the presence of a novel rather than routine failure. Ideally, the displays that result from a novel fault would be sufficiently different from the displays of routine faults. If the novel fault had a display different from routine faults, detection of a novel fault would seem to be assured. Unfortunately, no existing system has been designed from this perspective.

Another specialization is that novel fault diagnosis will occur when the operator has a problem space designed for routine operations and routine failures. It is not known if an existing problem space representation will interfere with novel fault diagnosis. It would seem difficult to believe that some interference does not occur.

A final distinction between novel fault diagnosis and most problem solving research has been how clearly the human can observe the system and the consequences of changes to it. For example, in cryptarithmetic, the human has complete information about the system, the legal operations, and their immediate consequences. Typically, when an operator controls a complex system, the system state is less clearly perceived, the available operations are larger in number, and their effects less clearly perceivable. The consequences of this imprecision are not well understood.

THE SYSTEM AND THE TASK

The Orbital Refueling System (ORS), a NASA-designed payload on the Space Shuttle, was selected for study [NASA 1985]. The function of the ORS is to refuel orbiting satellites with hydrazine, with the objective of extending their useful service life. As shown in Figure 1, the ORS fluid system contains a variety of components such as tanks, valves, pipes, etc. The operator controls the simulated ORS by opening and closing valves. Transfering fuel from propellant tank 1 to propellant tank 2 might proceed as follows. First, tank 2 pressure is reduced by momentarily opening valves 10, 11, 13, and 17. Second, tank 1 is pressurized by opening valves 1, 3, and 7. Gaseous nitrogen will flow out of the two small supply tanks, be pressure regulated, and fill tank 1 on one side of the bladder. To transfer fuel to tank 2, valves 5, 14, 15, 16, and 9 would be opened. Because this version of the ORS was for demonstration purposes, all transfers take place between the two large tanks rather than to a satellite fuel tank. There are several assemblies whose purpose was not explained in the above example. The relief valves RV1 and RV2 serve as a safety pressure relief. Check valve CV1 prevents backflow into the gas system. The bladders in tank 1 and 2 serve to isolate the fuel from the propellant and also to contain the fuel in the weightlessness of space. Some components (e.g., valves 10 and 11) may seem redundant; they are so by design for two failure tolerance.

The Diagnosis Task

The operator's task is to diagnose the failure in the system. This requires the operator to manipulate and observe the system, because a diagnosis cannot be determined uniquely from an observation of a state vector at a single point in time. A solution is an assignment of states to components such that the assignment's behavior is always identical to system behavior. For a single valve failure, the solution would be a normal state for all components save the failed valve, which might be jammed shut. The diagnosis problem can be viewed as a combinatorial search for a state assignment. The search is constrained by the laws of component physics. That is, a state assignment to a component imposes constraints on its neighboring components. For example, if a valve is opened and permits a flow down a pipe, the component receiving the flow must be in a state to accept the flow.

QUALITATIVE MODELS OF CONTINUOUS PHYSICAL PROCESSES

This section describes qualitative models: representations, the computational problems solved, and the specific needs of our aid of the qualitative model.

A qualitative model is a symbolic representation of a system. Its most basic description is of a component. A component is described in terms of its connections to other components and its behavior. Behavior is described in terms of the physical variables which are present at its connections. The differentiation between the structural description (connections) and the behavioral description is particularly important for insuring the robustness of a qualitative model. The isolation of each component in the behavioral description has usually been emphasized by other qualitative modeling [De Kleer and Brown 1983]. Our qualitative model represents the system at both the component level and at an aggregated level as paths. The motivation for this is the belief that a multi-level description is closer to the operator's internal model of the process.

From a given state, the behavior of a component is described in terms of the physical variables present at its ports. A physical variable (and its time derivative) may take several values. The time derivative usually has only one of three possible values: negative, zero, or positive. The variable itself may take either nominal or ordinal values. The nominal values usually correspond to points at which behavior (component or material) changes. For example, water temperature would have nominal values at freezing and boiling. Variables may also take on ordinal values (or relationships). For example, water temperature could be taken to be greater than freezing and less than boiling.

The nominal and ordinal values taken by physical variables are said to occur in a quantity space [Forbus 1984, Kuipers 1984]. The quantity space is a partial ordering on the physical variable values it contains. The partial ordering occurs because not all comparisons are relevant to understanding the physical system qualitatively. For example, consider a valve between two tanks, A and B. When the valve is opened, the resulting behavior is determined by the pressures in two tanks. The pressure at other unconnected points in the system is unrelated to the above behavior.

One question that is often raised is why bother with qualitative models. They are not, as it turns out, particularly fast or accurate. For engineering purposes they are inferior to analytic or numerical models. The answer to this question is, first, that the aid does not require a qualitative model; any system model will be acceptable if it can provide the required information to the operator. Our motivation for using a qualitative model is to test the hypothesis that humans use such models internally. Obviously, it is difficult to test this hypothesis directly. A weak test would be whether the qualitative model really aids human performance as described here. A stronger test would be finding similar reasoning weaknesses. As mentioned earlier, a qualitative model cannot answer some questions. If well-trained operators could not answer such questions, did not ask such questions, or could not use answers to such questions, there would be evidence for the hypothesis.

AN EXPLORATORY EXPERIMENT

An exploratory experiment was conducted to observe the strategies subjects used to diagnose the ORS. Three Georgia Tech students were used as subjects. The use of college students is usually considered a compromise in experimental research. Since some space shuttle astronauts have been engineers, this compromise is reasonable in this situation.

The training contained both theoretical and practical elements. First, the basics of gas and fluid transfer were reviewed. Second, there was an explanation of the normal and malfunction behavior of each component. Third, subjects were told how to test for a failed component and how to operate the system.

The subjects then solved five single failure malfunctions. The failures were as follows:

(1) Valve 13 leaked, allowing an unexpected pressure drop.

(2) Pressure transducer 2 was biased high.

(3) A leak to the environment developed between valve 10 and 11.

(4) The relief valve was open during a fuel transfer.

(5) Valve 8 leaked.

The data collected included a time-stamped record of the ORS commands issued and a tape recording of the subject's verbal protocols. The time to solution is shown in Table 1.

Subject / Problem	A	B	C
1	28.6	14.4	31.1
2	*13.8	*21.9	3.6
3	13.4	7.9	6.2
4	12.7	10.0	*21.9
5	7.5	8.3	12.3

Table 1. Time to solution. * denotes giving up.

A Post-hoc Analysis of Performance Data

The data from our preliminary experiment suggest several interesting characteristics of human diagnosis behavior, and which in turn suggested some directions for computer aiding. First, the time spent for a successful diagnosis is strongly related with the number of information gathering actions (IGA) ($r = 0.79$) and the average time between actions ($r = 0.77$). The latter two variables were not strongly correlated ($r = 0.21$). The implication of this is reducing the number of information gathering actions (IGA) is an important goal for improving diagnostic performance.

Second, we classified IGA's into effective ones (EIGA), which reduced the size of feasible hypothesis set, and ineffective ones (IIGA), which did not. We found that the number of EIGA is invariant among subjects and is also not significantly correlated with the total number of IGA. The total number of IGA is correlated with IIGA (corr.= 0.98), which outnumbered EIGA by 2.5 : 1. This suggests that a problem is solved by collecting the right number of EIGA (largely determined by the complexity of the problem). A better performance is possible when the effective actions are executed earlier in the diagnosis.

Third, we investigated how well the subjects detect the abnormal behavior of the system. We assessed the delay in diagnosis due to failures to collect information that would have revealed the abnormal system behavior. The delay showed high correlation ($r = .79$) with the number of ineffective actions. Also, 75% of effective actions were of abnormal behavior, and the remaining 25% were of normal behavior (negative evidence). Observations on abnormal behavior, if they are correctly interpreted, became effective actions in almost all cases. Thus, abnormal behavior of the system is probably the most important source of effective information.

The conclusion is that, to help the diagnosis, the cues for effective actions need to be given. Abnormal system behavior is worth watching for this purpose. When desiging an aid, a major advantage of using abnormal behavior is that inferring or requesting the human's current hypothesis is not necessary.

Observation of Strategies

There appeared to be three strategies that subjects used: hypothesis-driven evaluation, data-driven evaluation, and topographic search. Hypothesis-driven evaluation starts with the planning of a test procedure for a given hypothesis. The hypothesis needs to be explicit enough to enable the prediction of its resulting system behavior. A test plan would be diagnostic if, given that the hypothesis is true, the response of the system to the test is unique to the hypothesis. When a sufficiently diagnostic test has been planned, the test is executed and its result evaluated. This evaluation tends to be short because it has already been determined what the results might be.

With data-driven evaluation, the subject first examines a piece of data to determine if it is worth closer attention. This examination is done by comparing the data to expected system behavior. If the data turns out to be unexpected (i.e., not explained in terms of previously observed symptoms or normal behavior), then hypotheses are formulated to explain the data. Whether the formulation is successful or not, this piece of data is remembered by the diagnoses as another symptom to be used later during diagnosis.

Topographic search seems to help reduce the mental workload in diagnosis. Both above evaluation strategies involve deep reasoning with functional causalities. With deep reasoning, the former deduces necessary data from a given hypothesis while the latter formulate and evaluate hypotheses from the given data. Topographic search [Rasmussen 1984], without such a deeply based hypothesis, is used to find data. For instance, the sensor near the suspected component are read in hope that the reading may give some diagnostic information. An example of topographic search of hypotheses is suspecting nearby components when a sensor reading is out of the normal range. The differentiation of a single general hypothesis to several more specific hypotheses can be considered as topographic search.

Although it is not relevant to our diagnostic task, other forms of rules may be used as alternative ways of causal search. With experience or specific system knowledge, it is possible to connect a hypothesis with data through function-based reasoning [Rasmussen 1984].

AIDING WITH A QUALITATIVE MODEL

This section describes how the qualitative model is used as a foundation for aiding. For simplicity, the interface will be used to organize the presentation. The interface has four windows: schematic, interaction, sensor display, and hypotheses (Figure 2). Each window will be described first. The types of aiding that occur within the window will then be described. Finally, the justification for the aid, which is the human decision-making suboptimality we hope to mitigate, will be presented. It is possible that a form of aiding and a justification for aiding may apply to more than one window.

<u>Schematic Window</u>

The schematic window displays a schematic diagram of the ORS. The schematic always shows the commanded state of the valves. One form of aiding employed here is the set of components that should be at equal pressure given the commanded valve positions. Whenever the operator opens or closes a valve, the display changes the path to show this property.

The motivation for this is that the operator frequently makes a test among a set of components that should be at equal pressure. It should be noted that the qualitative model uses this same information internally in its simulation of the ORS. A related form of topographic information is flow paths, which are paths that should contain flow if the valves obey their commands.

Both of these forms of aiding support the operator during topographic search [Rasmussen 1985]. From a cognitive standpoint, both aids should lessen working memory loads. It is by no means difficult to determine equal pressure and flow paths without the aid, but it is extra work for the operator to do so.

The second aid and perhaps the most interesting is the what- if model with which the operator may test a hypothesis. The what-if model is a model that is parallel to the system model. The component states of the what-if model are set by the operator. Recall that the diagnosis task is to determine the states of the system components. The operator may use the what-if model to test a hypothesis. For example, suppose valve 13 is hypothesized to be leaking. Then, the operator may turn on the what-if model, set its valve 13 to leaking and all other components to normal. When activated, the behavior of the what-if model and simulation are displayed in parallel. The system can be put through a series of state changes to determine if the two behaviors are equal.

The motivation for this aid is to help the operator's mental model of the system. There are two ways this might help. First, the operator may have an incorrect or incomplete mental model. Second, the operator may have difficulty integrating correct component behavior to correct system behavior because of working memory limitations. In either case, the what-if model serves as a substitute for the operator's model. This does not mean that the operator need not understand the system at all; he or she must still set the component state. It also does not mean that the operator may not have trouble using this aid. We will return to this question later.

Interaction Window

The interaction window is where the operator's commands are echoed by the interface. The commands available to the operator include the following:

(1) Opening and closing valves.

(2) Comparing two pressures. On a real physical system, the numerical pressure could be displayed on the schematic. When a qualitative model is used, there is no scale in general to which a pressure can be referred. Instead, a pressure can be referred to other pressures in the system by the relations less-than, equal-to, or greater-than.

(3) Display of the first derivative of pressure (positive, zero, or negative).

(4) Turning the what-if model on and off.

(5) Making state assumptions in the what-if model.

When the what-if model is on, the open, close, and comparison commands apply both to the system and the what-if model.

Sensor Display

The sensor display contains the output from the comparison command: the relationship between two pressures or the first derivative of a pressure. The what-if model, if activated, has its corresponding output displayed side-by-side with the system model.

The aiding that occurs through this window is to indicate which observed behaviors deviate from normal behavior of the system. The aid runs a normal model (that is, a qualitative model with all component states normal) and compares its behavior to the system's behavior. Differences are highlighted. This display differs from conventional warning systems (for example, annunciator panels in nuclear power plants) in that reference is made to a system model, not a fixed point.

The strategy supported by this display is data-driven search, which was observed in our preliminary experiment. In the initial stages of diagnosis, the operator did not have a specific hypothesis. Instead, he or she collected data to develop one. The purpose of this aiding feature is to direct the operator toward more relevant data.

The human decision-making biases that we hope to mitigate all deal with suboptimal use of data or cues. Human have a limited ability to integrate more than three sources of information. Further, humans sometimes use irrelevant data, especially if it is salient. This display attempts to mitigate this by making important differences salient. Another deficiency of humans is a narrow focus of attention. The aid should work against this by displaying all differences, not just those on which the operator has focused.

Hypotheses Window

The hypotheses window will display a set of hypotheses that might be the cause of the observed symptoms. These hypotheses are simply state assignments to components (e.g., valve 13: leaking). The hypotheses will be listed in order of plausibility, according to a heuristic of symptom covering.

Many decision-making biases exist with respect to hypotheses. The one that is directly addressed is the difficulty humans have in generating a complete set of hypotheses [Mehle 1982].

Representativeness, anchoring, and confirmation bias often occur when humans select and evaluate biases. Representativeness refers to the tendency to select hypotheses that are easily recalled from memory. This could be due either to recent use of the hypothesis or to a close match between actual symptoms and symptoms covered by the hypothesis. Anchoring refers to the tendency to stay with an initial hypothesis even after it has been disconfirmed. Confirmation bias is the tendency to test data that will only confirm a hypothesis. It is in effect a failure to seek negative evidence. To mitigate these biases requires meta-aiding, as described below.

Meta-aiding

Earlier, we mentioned that the operator may have difficulty using the what-if model. Recall that the operator must make assumptions about the states of components. Having a what-if model means the evaluation of assumptions is easy, but making assumptions is not aided by the what-if model.

Meta-aiding is aiding the use of the what-if model--specifically, helping the operator choose component state hypotheses. While listing these hypotheses in the hypotheses window is an aid, it may be necessary for the interface to take a more active role. If anchoring and confirmation bias occur, it will be necessary for the interface to determine when the operator's hypothesis (expressed in the what-if model states) is no longer valid. When this occurs, the interface will step in to warn the user of his or her mistake.

CONCLUSION

An aid has been described for novel fault diagnosis in complex systems. To the best of our knowledge, this aid is unique in the following ways. First, the emphasis is on novel rather than routine faults. Second, it contains a qualitative model that may correspond to the human's internal model of the system. This model represents knowledge only of how the system works. Many of the proposed aiding schemes are proceduralized fault finders: they tell the operator what action to take. Third, the qualitative model is the basis for much of the aiding that takes place. Fourth, the interface specifically attempts to mitigate some human decision-making suboptimalities during fault diagnosis.

The current status of this aid is as follows. The aiding software for topographic path displays, flow paths, and the what-if model have been implemented. Hypothesis generation and the corresponding suboptimality detection have not. We feel it is premature to implement suboptimality detection (i.e., meta-aiding) without some experience with aiding by topographic displays and the what-if model.

ACKNOWLEDGMENT

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

REFERENCES

Davis, R., "Diagnostic reasoning based on structure and behavior," Artificial Intelligence, Vol. 24, pp. 347-410, 1984.

De Kleer, J. and Brown, J.S., "Assumptions and ambiguities in mechanistic mental models," in D. Gentner and A.L. Stevens (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Forbus, K., "Qualitative process theory," Artificial Intelligence, Vol. 24, pp. 85-168, 1984.

Gentner, D. and Stevens, A.L. (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Kuipers, B., "Commonsense reasoning about causality: Deriving behavior from structure," Artificial Intelligence, Vol. 24, pp. 169-203, 1984.

Mehle, T., "Hypothesis generation in an automotive malfunction inference task," Acta Psychologica, Vol. 52, pp. 87-106, 1982.

Miller, R.A., Pople, H.E. Jr., and Myers, J.D., "INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine," in Readings in Medical Artificial Intelligence, Clancey, W.J. and Shortliffe, E.H. (eds.), Reading, MA: Addison-Wesley, 1984.

Morris, N.M. and Rouse, W.B., "The effects of type of knowledge upon human problem solving in a process control task," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-15, No. 6, 1985.

Morris, N.M. and Rouse, W.B., "Review and evaluation of empirical research in troubleshooting," Human Factors, Vol. 27, No. 5, October 1985.

NASA Johnson Space Center, "Orbital refueling demonstration system description," Program Development Office, October 21, 1985.

Newell, A. and Simon, H.A., Human Problem Solving, Englewood Cliffs, NJ: Prentice-Hall, 1972.

Rasmussen, J., "Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, pp. 257-266, 1983.

Rasmussen, J., "Strategies of state identification and diagnosis in supervisory control tasks, and design of computer based support systems," in W. Rouse (Ed.), Advances in man-machine systems research, Vol. 1, JAI Press, 1984.

Rasmussen, J., "The role of hierarchical knowledge representation in decision making and system management," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-15, No. 2, pp. 234-243, 1985.

Rouse, W.B. and Morris, N.M., "On looking into the black box: Prospects and limits in the search for mental models," to appear in Psychological Bulletin, 1986.

Shepherd, A., Marshall, E.C., Turner, A., and Duncan, K.D., "Diagnosis of plant failure from a control panel: A comparison of three training methods," Ergonomics, Vol. 20, No. 4, pp. 347-361, 1977.

Shortliffe, E.H., Computer-Based Medical Consultation: MYCIN, New York: American Elsevier, 1976.

Wickens, C.D., Engineering Psychology and Human Performance, Columbus, OH: Charles E. Merrill, 1984.

Wohl, J.G. "Maintainability prediction revisited: Diagnostic behavior, system complexity, and repair time," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-12, No. 3, 1982.

Wohl, J.G. "Cognitive capability versus system complexity in electronic maintenance," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-13, No. 4, 1983.

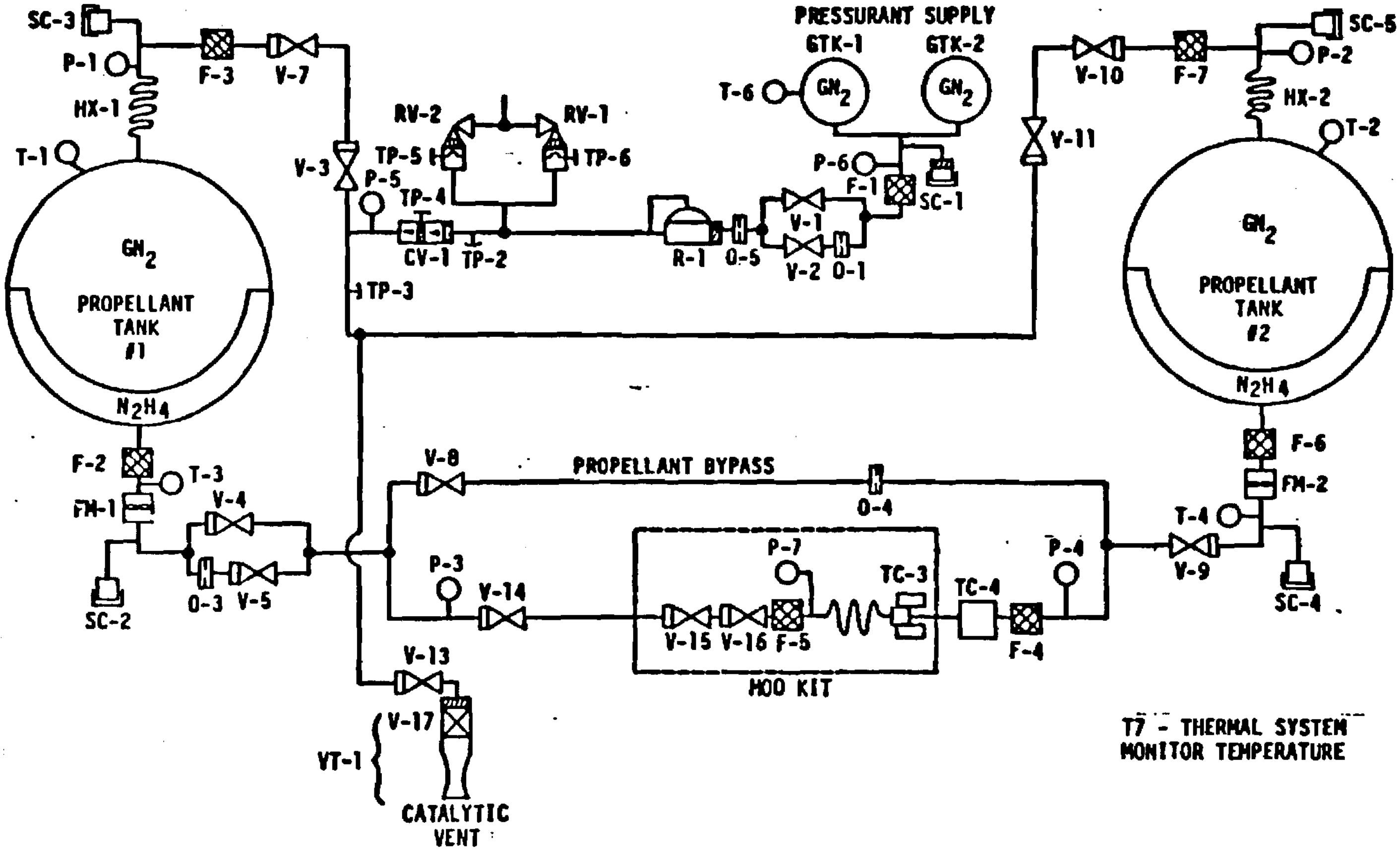


Figure 1. The Orbital Refueling System.

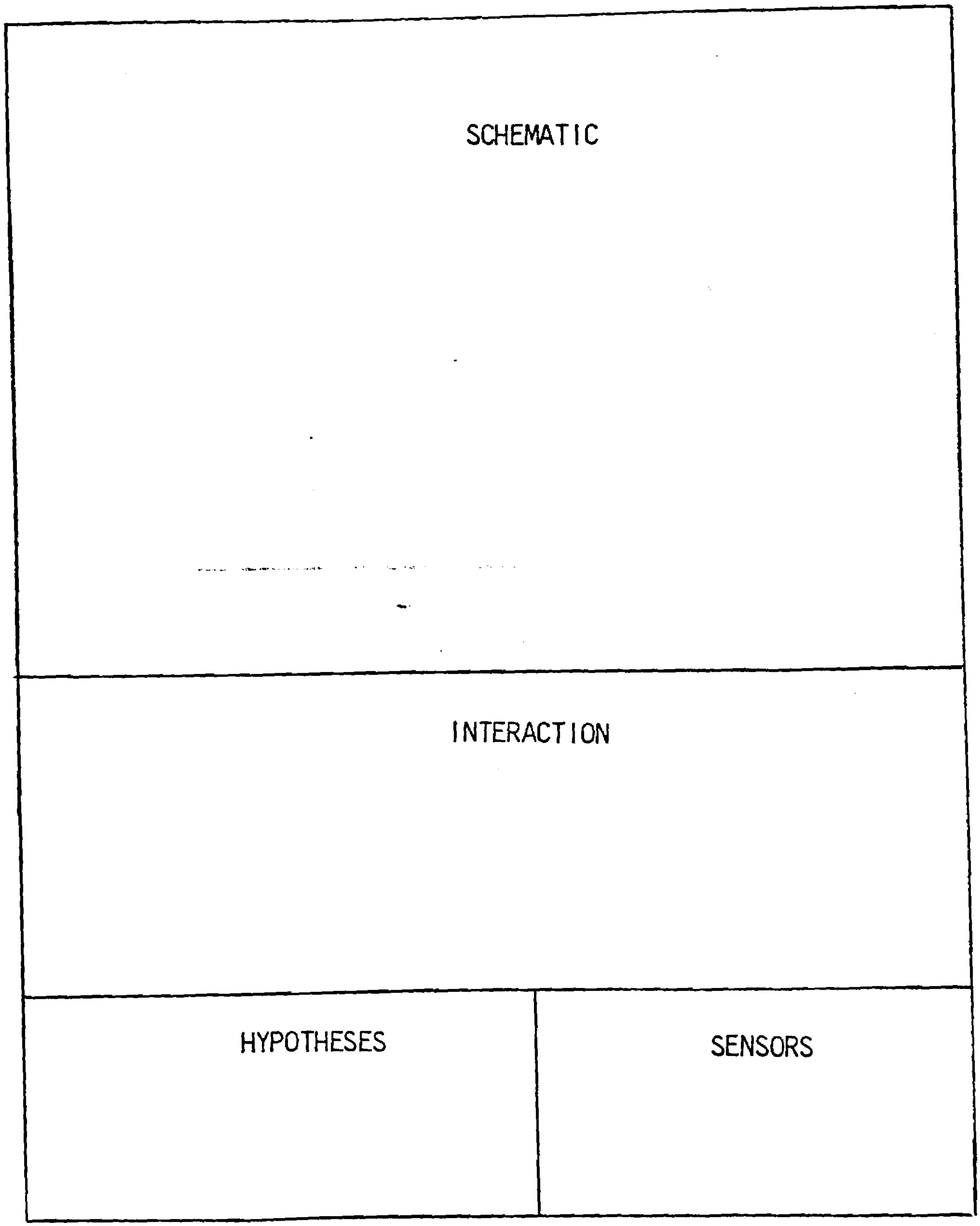


Figure 2. The operator's display.

C

```
                      C36
 <-VT----V17-------V13                   RV      C33      C32          C31    /""""""\
                     | C35                 |              .-----V1---.     | GTK  |
    .--V7----V3--------------< CV <-------REG--05--:              :---\______/
    |    C37       |      |           C34              `---V2--01-'         |
    |C38           |      @ P5                                              @ P6
    |              |                         C39            C40
 /""""""""""\~@ P1  `-----------------V11------V10-------------/""""""""""\~@ P2
 | TK1G/L  |                                                   | TK2G/L  |
 \_________/                                                   \_________/
    |                         @ P3                 @ P7     @ P4         |
    | C1                      |      C6     C7     |        |       C10|
    |        .-----V4---. C4 .----V14---V15---V16-->TC<-----.          |
    `-------:           :----|                      C8        C9|----V9--'
             `-03----V5-'    `-------V8-------04----------'
                                         C5
```

Exercise 1.
    Situation: P1 is found too low and still decreasing.
    Fault: V13 leak
    valves open: V3, V7, V10, V17 / V4, V14, V16, V9

Exercise 2.
    Situation: P2 appears to be too high.
    Fault: P2 high bias
    valves open: V1, V3, V10, V17 / V5, V15, V9

Exercise 3. (confg-1)
    Situation: P1 is low and decreasing.
    Fault: pipe leak between V10 and V11 (c39)
    valves open: V3, V7, V13, V11, / V4, V14, V15, V9

Problem 1.
    Situation: During a fuel transfer TK1L -> TK2L,
                   P2 does not increase.
    Fault: V5 fail closed
    valves open: V3, V10, V11, V17 / V5, V14, V15, V16, V9

Problem 2.
    Situation: P2 is too high. V11 was found leaking, but
                   there is one more anomaly.
    Fault: V7 failed open
    valves open: V3, V13, V10 / V4, V14, V16, V9

```
<-VT----V17-------V13                  RV                                /""""""\
                   |                    |               .-----V1---.    | GTK |
 .--V7----V3---------------< CV <-------REG--05--:                 :---\_____/
 |          |        |                                 `---V2--01-'       |
 |          |        @ P5                                                 @ P6
 |          |
/""""""""""\~@ P1  `----------------V11------V10-------------/""""""""""\~@ P2
| TK1G/L |                                                  | TK2G/L |
\________/                                                  \________/
 |                     @ P3          @ P7    @ P4                  |
 |                     |             |       |                     |
 |        .-----V4---.   .----V14---V15---V16-->TC<-----.          |
 `-------:          :----|                              |----V9--'
          `-03----V5-'   `-------V8--------04-----------'
```

Problem 3.
    Situation: P2 is lower than it shoud be.
    Fault: V8 leak (while P2 > P1)
    valves open: V3, V7, V10, V17 / V4, V15, V16, V9

Problem 4. (confg-2)
    Situation: During a fuel transfer TK1L -> TK2L,
    P2 increases too slow.
    Fault: pipe between V3 and V7 (C37) leaks
    valves open: V3, V13, V11, V10 / V5, V14, V16, V8, V9

Problem 5.
    Situation: a gas transfer GTK -> TK2G proceeds too slow.
    Fault: V15 leak
    valves open: V3, V17, V1, V11, V10 / V4, V14, V16, V9

Problem 6.
    Situation: During a gas transfer from GTK to TK1G,
    P1 increases too slow.
    Fault: RV leak
    valves open: V1, V3, V7, V10, V13 / V4, V14, V9

(B)

INSTRUCTIONS - Part 1.

TIME (:   )

I. The Orbital Refueling System (ORS)

The purpose of the ORS is to refuel orbiting satellites on their orbits. As shown in Figure 1 (in the separate sheet), the ORS fluid system contains various components such as tanks, valves, pipes, etc. Because this version of the ORS was for demonstration purposes, all transfers take place between the two tanks rather than to a satellite.

Let's look at the components in the schematic. First, 'XX' and '==' indicate closed and open valves respectively. The operator controls the ORS only by opening and closing valves. For example, You can open/close V3 by the commands OP V3 and CL V3.

There are 4 orifices: namely O1, O3, O4, and O5. Find them in the Figure. An orifice is a designed source of resistance. When there is a mass flow through an orifice, there is a pressure reduction across it. Dropping pressure through orifices is at times useful to control the flow rate. Also, O1 and O5 reduce pressure to the regulator.

Now find GTK, which stands for the Gas TanK. This tank contains high pressure nitrogen gas. Find Tank1 (TK1G and TK1L) and Tank2 (TK2G and TK2L) too. They are the fuel tanks. TK1G is the gas part of Tank1, which is separated by a flexible diaphragm from the liquid part (TK1L) of the tank. The two parts always share the same pressure.

On the path from GTK to TK1G, you will find 'REG' (REGulator) and 'CV' (Check Valve). The regulator produces a constant output gas pressure even though the input pressure varies. The check valve allows the gas to flow forward only (i.e., right to left).

Find 'RV'. It stands for a Relief Valve. If the pressure goes up beyond some dangerous level, the relief valve will automatically open to decrease the pressure. The operator can also manually open/close the 'RV' as any ordinary valves by OP RV or CL RV. At the top left, you see 'VT', which stands for VenT. You may release pressurizing gas through the vent by opening V13 and (___).

The lower half of the schematic (from 'TK1L' to 'TK2L') is the liquid (fuel) part. There, 'TC' is for Terminal Coupling and is assumed always being connected during our diagnostic missions.

To transfer fuel from 'TK1L' to 'TK2L', Tank1 needs to be first pressurized by opening valves betwen GTK and TK1G. In the above schematic, TK1G is being pressurized by the gas through the open valves (___), (___), and (___). Since TK1L has always the same pressure as TK1G, it is being pressurized too. Then, the gas flow may be stopped by CL V2. The fuel may be transferred by opening valves between the two tanks. In the above, the operator would simply open (___), hence issue a command (________) to do this. The tank of higher pressure will become the source and the other will receive the fuel.

The following is important. There are seven pressure sensors (P1 to p7) in the ORS, (__),(__),(__), and (__) in the gas part, and (__),(__), and (__) in the liquid part. To read them, you have only two commands:

D P1

: to see the 'D'erivative of P1.

Answer: + for P1 increasing, - decreasing, and = constant.

C P2 P4

: to 'C'ompare P2 and P4.

Answer: > when P2 > P4, < when P2 < P4, and = when equal.

The command D is valid only for tank pressures, namely, P1, (___), and (___). In pipes, unlike the tanks which have considerable capacity, the pressure change is instantaneous so that you can't expect to see + or - as the answer to D P5. D IS MOSTLY USED TO CHECK IF THERE IS A FLOW FROM/INTO A TANK.

As the gas or liquid flows from one tank to another, its pressure decreases along the path. A pressure drop can only occur across a resistance. When the fluid passes an orifice, which has significant resistance, the pressure will decrease. An abrupt change in the conduit shape, such as from a pipe to a tank or vice versa, also produces resistance and results in a pressure drop. We will assume that pipes or valves normally have negligible resistance. C is the command which is frequently used to check the pressure drop along the path.

Another use of C command is to check if two sensors, which are supposed to be equal, agree with each other. When two or more sensors are connected by an open path, and if there is no flow through the path between the sensors, they all should read the same pressure. Resistance doesn't matter when there is no flow. If the sensors read differently, either there actually is a flow in the path (e.g., due to a leaky valve or pipe) or at least one of the sensors is wrong.

Keeping this in mind, you are now able to predict how the sensors will behave when you open or close valves. Three situations are summarized here.

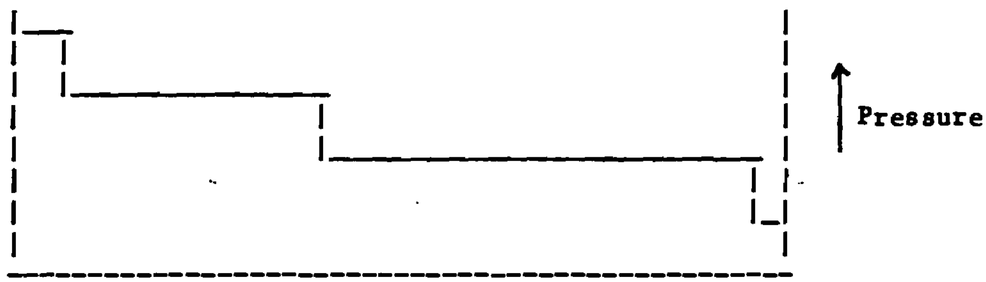
1. When a flow exists.

   a. The pressure decreases in the source and increases in the sink.

b. The pressure drops by resistance while the material travels along the path. This is shown in the following diagram.

```
TANK 1 -------------01--------==12-----------TANK 2
(source)                                      (sink)
```

Flow ----->



In Figure 1, there is a gas flow from GTK to TK1G.

Predict the results:

D P6 --> (__)

D P1 --> (__)

C P6 P5 --> (__)

C P5 P1 --> (__)

Now, how about

C P1 P3 --> (__) ?

Did you consider that an orifice reduces pressure only when there is a flow through it? If not, check your answer again.

Now, suppose P2 > P1 and V8 is open.

Then the fuel will flow from (____) to (____) and:

C P2 P4 --> (__)

C P4 p7 --> (__)

C P4 P3 --> (__)

C P3 P1 --> (__)

2. In case that there is no flow (V8 is closed again).

In Figure 1, there is no flow from or to 'TK2G' and 'TK2L'.

All the pipes around the tank will share the same pressure.

Thus,

C P2 (__) --> =

C P2 (__) --> =

If you close V9, the pressures on both sides (will, will not) change. Therefore,

C P2 P7 --> (__)

On the other hand, when you close v3, you expect

C P5 (__) --> =

3. Special case of 2.

In Figure 1, suppose V17 is leaking.

It is open to the environment which has zero pressure.

Even though the operator closes V3, the gas will continue to flow from (_____) to the environment. Thus,

D (__) --> -

C P5 P6 --> (__)

C P5 0 --> (__)

If the operator closes V2, Since the pipes do not have significant capacity, the gas escapes right away. Therefore, immediately after closing V2, you get

C P5 0 --> (__)

** We assume the capacity of components (except tanks) to be always negligible however small the size of a leak may be.

The same will be (true, false) when V17 is closed but the pipe between V13 and V17 leaks.

TIME (:)

BEFORE YOU START NEXT PART, RETURN THIS PART TO THE EXPERIMENTER.

TIME (:   )

II. Malfunctions

We will now discuss possible malfunctions for each component.

1. valve (including check valve and relief valve)

   a. leak - in spite of being commanded to be ´closed´, it allows some, though not a full, flow. There is a resistance when commanded closed. When commanded open, it acts normally.

   b. fail open - no matter what you command, it remains fully open.

   c. fail closed - no matter what you command, it remains closed.

2. regulator

   a. fail open - always remains fully open without reducing the pressure.

   b. fail closed - always remains closed whatever the input pressure is. (No gas passes through the regulator.)

3. orifice

   a. fail open - fails to provide resistance or pressure drop, allowing the material to flow freely.

   b. fail closed - prohibits flow.

4. conduit (including ´TC´, the terminal coupling)

   a. leak - leaks gas or liquid to environmental space. (remember that a valve leak is THROUGH the valve, not to the environment)

   b. fail closed - completely prohibits flow.

5. vent

Since the ´VT´ is a simple opening to external space, its working and malfunctioning is the same as a conduit.

6. sensor

a. biased high - reports a higher pressure than the actual one.

b. biased low - reports a lower pressure than the actual one.

c. dead - fails to follow the change of pressure, reading 0 or other fixed pressure.

TIME (:   )

III. Commands

We will summarize the commands you can use. There are only two commands for operation -- OP and CL. You can open or close only the valves. Examples are

OP V3

CL V17

There are two commands, 'C' and 'D', to get information about pressure through the sensors. Followings are the examples.

C P1 P3

D P2

C P5 0

The last command compares P5 with 0, the environment pressure of outer space.

** Now, call the experimenter. You may ask him any questions.

TIME (:   )

IV. An example of ORS operation.

** You need to use the terminal for this section. The experimenter will help you through this section.

Now, you will undertake a very typical operation as an exercise. Also, through this example, you can become more familiar with the commands. Simply follow the steps one by one with care. Don't open/close the valves otherwise, although you can freely read any sensors at any time.

a. type: (EXERO) and hit 'return'.

The familiar schematic now appears on the top half of the screen. Notice that the symbol 'XX' indicates a closed valve, and '==' an open valve. The symbol shows so-called 'commanded' position. The actual position can be different from this switch position when a valve malfunctions.

The fuel needs to be transferred from 'TK1L' to 'TK2L'. To achieve this transfer, the pressure in 'TK1L' should be higher than that of 'TK2L'. So, let's pressurize the source tank by providing high pressure from GTK

Please write in your answers whenever you are asked.

b. type: OP V1

What happens? (When you are asked like this, write down your guess on the system behavior resulted by the command.)

Try to confirm the above answer by observing sensors. Then, give a set of commands (including at least a 'C' command) that are useful for this.

c. type: CL V1

What happens?

How do you confirm it? (answer as in b.)

d. type: OP V8

What happens?

How do you confirm it? (answer as in b.)

Check 'C P3 P4', 'C P4 P7' and 'C P1 P5'. Can you explain them?

e. type: CL V8

OP V16

Check 'C P3 P4'. Can you explain it?

e. type: CL V4

Give all the sets of equal pressure sensors.

TIME (:)

Congratulations! Your first mission has successfully been completed.

INSTRUCTIONS - Part 2.

Before you start, please review Part 1 again. Especially, you need to be familiar with sections II and III of Part 1.

I. Diagnoses

The followings are examples of typical diagnostic procedures. Following the reasoning, fill in the parentheses.

a. To check a sensor

See Figure 1. Suppose you want to check the sensor P3. You can close V3 and expect (___) to read the same as P3. If not, P3 probably is bad. Of course, the bad one may be (___) rather than P3. To check further, you can close V9, open (___), and compare P3 to P4 or (___).

b. To check a conduit leak (to environment)

If there is a leak between V4 and V14, D P1 can either give - or + depending on whether the input flow rate to TK1G is greater than the output rate from TK1L. If you close V7, a leak between V4 and V14 will cause a decrease in the sensor (___). But, when the valve (___) is closed, TK1L will stop loosing the pressure. This means that the leak is in the { left, right } hand side of the valve. Another evidence of a leak between V4 and V14 is that C P3 O ---> = { before, after } you close V5.

c. To check a valve leak

Suppose you found D P2 gave +. This is possible if one of (___) and (___) is leaking. You may suspect that even two valves (___) and (___) failed together. If you close V10 and find the flow stopping, which makes (_____) return =, you have the evidence that the flow was from (____) and the leaky valve was (___).

If closing V10 does not stop the flow, you will first suspect (___) since one valve failure is more likely than a two valve failure. If closing V5 or V9 results in D P2 ---> =, the problem is in the { gas, liquid } part. Now, after you open V5 or V9 again, if closing V16 stops the flow, then the flow was through { V8, V14 and V15 }.

Now, let us consider several situations to see how you can test your hypotheses. You will be given a hypothesis for each problem. Each hypothesis implies that only one component is suspected. Prove or disprove the hypothesis.

TIME: (:   )

1. Hypothesis: the pipe between V13 and V17 leaks.

   Type (HYP01) and start when the diagram appears.

2. Hypothesis: V11 leaks.

   Type (HYP02) and start when the diagram appears.

3. Hypothesis: V2 failed closed.

   Type (HYP03) and start when the diagram appears.

4. Hypothesis: CV failed open.

   (Hint: you can open/close RV as well as other Valves.)

   Type (HYP04) and start when the diagram appears.

5. Hypothesis: P2 is biased high.

   Type (HYP05) and start when the diagram appears.

TIME: (:   )

II. Exercises

When you are diagnosing the ORS, you will be introduced to a malfunction situation and given the symptoms so far identified. The previous operation was being done by another personnel. Your mission is to diagnose the system and find out the anomaly AS PRECISELY AS POSSIBLE so that another crew could easily fix it. For example, if you suspect a valve leak, you have to continue until you can say which valve it is. A conduit malfunction can be traced down to ′between valve a and valve b′, where valves include the check valve (CV).

You have to THINK ALOUD during the diagnosis. That means, you should utter everything that arises in your mind or in action. DON′T try to EXPLAIN what you HAVE thought; speak out WHILE you are THINKING. Speaking must not be an extra work. You don′t have to give complete or composed sentences. The components which have names on the schematic may best be called by the names. Others, mostly pipes, may easily be called ′right to′ or ′left to′ a named component. Again, please KEEP TALKING OUT. Speak everything that goes on in your mind regardless of its importance. Also, whatever you type in on the keyboard needs to be spoken out. If you stop speaking for any length of time, the experimenter will prompt you with "What are you thinking?"

Your performance is measured by the sum of time you spend for the problems; solve the problems in as little time as possible. However, give your answer only when you are completely convinced it is correct. And, don′t give up, at least easily. The penalty for a wrong answer is great; giving up, even greater.

Now, proceed with exercises 1 and 2.

RETHINKING EXPERIMENTAL PROCEDURE

Findings from the 1st Experiment (Testing N Feature)

1. With enough training, the problem complexity becomes the biggest source of variation.
2. Subject variation may be reduced as much as to a standard deviation of around 0.3 mean.
3. The training effect was examined using Time/IGA. It was quite stable and showed similar pattern from problem to problem among subjects.
4. No significant interaction between the training effect and the aiding effect or subject effect were indicated from the data.
5. The "N" feature did not show positive effects.

Refinement of the Training Procedure

1. More exercise (2~3) problems are needed for "warming-up" before the actual problems.
2. Clearer statements and no question for the 1st session and "solve-it-together" for the 2nd session.

Experimental Design

1. The constraint of having to give a problem to a subject only once restricts the possibility of a factorial design. No replication in the S X P cells leaves the two following designs.
2. Design 1 confounds Problem and Position.

   Design 2 is a Graeco-Latin design which separates Problem and Position.

	P1	P2	P3
S1	ua	0	0-N
S2	0	0-N	ua
S3	0-N	ua	0

Design 1.

	1	2	3
S1	P1,ua	P2,0	P3,0-N
S2	P3,0	P1,0-N	P2,ua
S3	P2,0-N	P3,ua	P1,0

Design 2.

3. Confounding Problem and Position

   - As long as the training effect is not correlated with the aiding effect, this design will not degrade the efficiency or validity of the experiment. (We try to minimize the training effect, anyway.)
   - Although the training effect is not measured separately, it is not an important purpose of this experiment.
   - This design allows freedom of replication and keeps the analysis relatively easy.

4. Graeco-Latin Design

   - The main advantage is that we may estimate the training effect. However, the training effect is closely related to the problems. There would be more learning from a difficult, hence long, problem. If such a problem comes first, more improvement will occur after the first session. This violates the no-interaction assumption in a Graeco-Latin Design. Not only the training effect will not be properly estimated, but also the efficiency of test will be degraded since the actual interaction will be merged to the error term.
   - Design 1 allows more flexibility of replication. 9X6 or 12X6 are possible replications with Design 1, but are not allowed in Design 2.

5. Conclusion

- If we are concerned with the Training effect, than we need to confound it with Problem since there may be a strong interaction between the two. If the Training effect is not so high (which is the likelier case as the data indicates), Design 1 is readily justified.
- To estimate the interaction between Problem and Aiding, we need replication with subjects for each treatment combinations. This leads to the following design (Winer, "Statistical Principles in Experimental Design", 1962).

	P1	P2	P3	P4	P5	P6
G1	-	0	0-N	-	0-N	0
G2	0	0-N	-	0	-	0-N
G3	0-N	-	0	0-N	0	-

In this plan, G1, G2, and G3 are groups of an equal number of subjects. If the interactions with the group factor are negligible (this assumption is reasonable if the groups represent random subsamples from a common population),
the following model will be appropriate for the analysis (Winer, 1962).

$$E\,[Y(ijkm)] = m + G(k) + S(k|m) + P(i) + A(j) + P.A(i,j)$$

where G(k) is the effects associated with groups and S(k|m) effects associated with subjects within the groups.

EXPERIMENTAL PROCEDURE

I. Purpose of the Experiment

There are diagnostic situations in which causal reasoning about the physical system plays a central role. Such situations may be created by a system failure that the operator has not experienced. The irrelevancy of previous experience prohibits a direct mapping from symptoms to causes. Also, the base rates for hypotheses are normally not available due to the lack of experience. As a result, the diagnosis will primarily be based on causal reasoning about the system.

Aiding based on a qualitative model of the system seems to deserve consideration because the human's causal reasoning is also claimed to be qualitative. The qualitative model will be able to predict and describe the system events which are believed to be important to human reasoning. This should cause the information produced by the model to be highly compatible with the human information processing.

One purpose of this experiment is to test the validity of this aiding approach. More detailed interest is in the relative effectiveness of different aiding information that can be provided by the model. In the next section, the experiment planned for this purpose is described. The design of experiment and the analysis of results are discussed in the last section.

II. The Experiment

This section begins with a brief description of the Orbital Refueling System (ORS), which is the context of problem solving in the experiments, and the interface. A more detailed discussion may be found in the previous papers

[Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics, pp.1222-1227; IEEE Transactions on Systems, Man, and Cybernetics, to appear] and the thesis proposal. Then, a description of the experiment in terms of problems, independent and dependent variables, subjects, and training will follow.

The ORS and the Interface

In the ORS as described in the thesis proposal, as in most plants, it is not possible to test each component directly. A diagnostic hypothesis can only be examined indirectly through testing operations. Because of this, the diagnosis of a novel failure in this system will more heavily rely on causal reasoning. This makes the ORS a good problem solving context for our experiment.

The ORS is qualitatively simulated on the center's Vax 11/780 computer. The interface has four windows (Figure 1). The schematic window shows a schematic diagram of the ORS. The commanded positions of valves are shown on

```
---------------------------------------------
|                                           |
|                                           |
|            S C H E M A T I C              |
|                                           |
|                                           |
|-------------------------------------------|
| I N T E R A C T I O N                     |
|-------------------------------------------|
|                    |                      |
|  S E N S O R S     | H Y P O T H E S E S  |
|                    |                      |
|                    |                      |
---------------------------------------------
```

Figure 1. The ORS Interface

the schematic. Below the schematic, the operator's commands are echoed in the interaction window. The operator can open/close valves, read the time derivative of a pressure sensor, and compare two pressure sensors. The output from the above sensor display commands is displayed in the sensor window. Under certain aiding conditions, suggested sensor readings will also be displayed in this window. The hypothesis window is used only with an aiding feature. It displays a set of hypotheses set by the operator. These hypotheses are simply state assignments to components.

Problems

For each problem, the subject is given a detected symptom and asked to diagnose the malfunction as precisely as possible. There may be one or two bad components. When two components are bad, the subject is told of one malfunction and is asked to find the other. The problems include valve leaks, pipe leaks, blocked valves, a check valve failure, a relief vent leak, and sensor failures.

Independent Variables

The effects of different aiding information will be examined. Each type of information correspondents to a hypothesized, model-based processing that the operator does during diagnosis. The first processing is called N, which is to predict the normal system behavior after a given operation. The second is O, which is to envision the actual system behavior from limited observation. The third is O-N, the difference between O and N, which is often crucial in an efficient search for the diagnosis. The last processing is called O-H, which calculates the discrepancies between the observed system response and the operator's hypothesis.

Dependent Variables

Many different performance measures were tried with our data from the pilot experiment. The number of information gathering actions (#IGA) appears to be a clear alternative to the time to solve (TIME). An information gathering action is judged to be effective when it reduces the size of feasible hypothesis set. To achieve this, an IGA should be able to remove at least one hypothesis from the feasible set. In addition, it must not be redundant with respect to the information so far collected. We have denoted the number of effective IGA's by #EIGA, and that of ineffective ones by #IIGA.

The pilot experiment showed that #IIGA is a good predictor of TIME ($r = 0.83$; $p < 0.01$). Although several other measures were examined with the data, they either turned out to have insufficient resolution or showed high correlations with the above measures. Thus, the aboves will be the most important measures in the main experiment. However, other measures will be collected for supplementary analysis. The measures are:

Time :	Time to solve the problem
#IGA :	Total number of Information Gathering Actions
#EIGA:	Number of Effective IGA
#IIGA:	Number of Ineffective IGA
#BT :	Number of Bad Tests of Good Hypotheses
#BH :	Number of Good Tests of Bad Hypotheses
#RT :	Number of Redundant Tests

Subjects

Eighteen to twenty four undergraduates in the ISyE 3010 class will serve as volunteer subjects. The subjects will receive extra credit for participating this experiment. They are motivated by giving different extra credit according to their performance: 7% for top one third of the subjects, 6% for the next one third, 5% for the rest.

Training

The goal of our training is to facilitate the subjects with correct causal reasoning about the ORS and reasonably stabilized diagnostic skills. However, if a subject is exposed to a kind of problem several times in a short period, the subject may develop diagnostic procedures that do not require causal reasoning. That means the problems become routine failures rather than novel ones to the subjects.

Two training sessions will prepare the subjects for the final experiment. Training session 1 starts with basic principles derived from fluid dynamics. Then, possible malfunctions for each component are discussed. Finally, the subjects will undertake a simulated ORS mission, during which envisioning of normal system response is practised. Session 2 teaches elementary diagnostic procedures such as checking a sensor bias or a valve leak. The subject then is required to plan testing procedures for five typical hypotheses. Each procedure will be discussed with the experimenter until the subject develops (and understands) a correct procedure. The subject then solve three real problems as exercises. Session 1 usually takes 1 to 1.5 hours. Session 2 is normally takes 2 hours, but varies depending on the subject's pace.

The performance of subject in the training sessions is closely monitored. The principles part contains many questions to ascertain proper understanding.

The answers are checked during the same session and, whenever necessary, discussed again. Problem solving exercises are also attended by the experimenter and necessary discussion or re-explanation is provided. The result is that initially poorer subjects will spend more time in training rather than end with poor understanding. Our experience is that by the end of the second session, subjects performed satisfactorily and showed little additional improvement in diagnostic skill.

III. Experimental Design

Rationale for Three Experiments

The features will be examined by three experiments. The display of aiding information constrains those features that can be tested together. A subject should not be exposed to both N and O features since severe interference is expected. This is because O and N information is displayed identically but has different meaning.

O-H and O-N for the same reason should not be used together. When O-H is used, it acts as O-N until the subject expresses one or more hypotheses. This makes a direct comparison between O-N and O-H difficult. Even if O-H really improves the performance, its contribution will be depend on the extent to which a subject uses it. Different performance criteria need to be used to evaluate the potential benefit of O-H. (The frequency of bad hypothesis testing (#BT) should be emphasized rather than time to solve (Time). The ratio of #EIGA and #IIGA with or without a hypothesis selected may also be compared. These comparisons need to be made against the O-N aiding condition.)

The above considerations led to the following three separate experiments.

1. Test of N against unaided situation
2. Test of O, O-N, against unaided situation
3. Test of O-H against O-N

Differences in the complexity of problems and differences between users are expected to introduce large variation in the performance. To enhance the efficiency of the experiment, a Latin square design which uses problem and subject as two blocking variables is desirable. The treatment levels will be counterbalanced for practice effects. Also, the Latin square design may be replicated to attain enough data points. This design is used for all three experiments. The ANOVA table for this design is given in Appendix A. The first experiment to evaluate the N feature is shown in Figure 2. Figure 3 shows the experiment for testing O and O-N features.

The above design does not estimate interactions because only first order effects are of interest. There is no hypothesis that corresponds to an

		PROBLEMS					
		P1	P2	P3	P4	P5	P6
SUBJECTS	S1	N	-	N	-	N	-
	S2	-	N	-	N	-	N
	S3	N	-	N	-	N	-
	S4	-	N	-	N	-	N
	S5	N	-	N	-	N	-
	S6	-	N	-	N	-	N

Figure 2. Latin Square Design for N effects in Experiment 1.

		PROBLEMS					
		P1	P2	P3	P4	P5	P6
	S1	-	0	0-N	-	0-N	0
	S2	0	0-N	-	0	-	0-N
SUBJECTS	S3	0-N	-	0	0-N	0	-
	S4	-	0-N	0	-	0	0-N
	S5	0	-	0-N	0	0-N	-
	S6	0-N	0	-	0-N	-	0

Figure 3. Latin Square Design for 0 and 0-N in Experiment 2.

interaction between 0 and 0-N.

Pairwise comparisons will be executed using procedures by Tukey, Bonferroni, and Scheffe [J. Neter and W. Wasserman, "Applied Linear Statistical Models", 1974, Irwin]. Since the sample size is balanced, the Tukey test can be used and is expected to be most sensitive.

In the third and final experiment, the 0-H option in the 0-N feature will be tested against 0-N feature only. As in the test for N, 6 subjects will be used for this analysis.

Appendix A.

Source	Sum of Square	d.o.f
Treatment		p-1
Problems		n(p-1)
Subjects		n(p-1)
Error		$n^2p^2-p-2n(p-1)$

Where,

p : Number of subjects, problems

n : Number of replications

ANOVA Table

for Replicated Latin Square Design without Interaction

D

```
;fuel.lsp
; physical modeling representation and manipulation
; John M. Hammer
; 3/10/87
;(component
;  (name ())
;  (type ())
;  (ports
;    (sf-list
;      (port
;       (type ())
;       (name ())
;       (pressure ())
;       (flow ())
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;      (port
;       (type ())
;       (name ())
;       (pressure ())
;       (flow ())
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;       );sf-list
;  (state-variables
;    (sf-list
;      (state-variable
;       (mass ())
;       )
;      )
;    )
   (parameters
     (sf-list
       (parameter
        (resistance ())
        (value ())
        )
       (parameter
        (volume ())
        (value ())
        )
   (behaviors
     (sf-list
       (behavior
        (cond (<expr>))
        (eqns (sf-list <ar> <ar> <ar>))
        )
       (behavior
```

```
;       (cond ((<expr>))
;       (eqns (sf-list <ar> <ar> <ar>))
;       )
;      );sf-list
;
; <expr>
; an expr is either an ar (defined below) or the and of
; a list of ars:
;       <expr>  ::= <ar>
;               ::= pand <ar> <ar> ... <ar>
; <ar>
; an algebraic relationship, which could be an equation or an
; inequality (possibly a constraint)
; examples:
;       a = 1           (peq a 1)
;       b < 3           (p< b 3)
;       x+y=z-q         (peq (p+ x y) (p- z q))
;
; in ports and parameters, there are slot names that are physical
; dimensions (e.g., resistance, pressure, flow)
; an example of a valve
;(component
;  (name (valve14))
;  (type (valve))
;  (ports
;    (sf-list
;      (port
;       (type (liquid))
        (name (in-port))
;       (pressure (in-pressure))
        (flow (flow))
;       (connection
          (tie-point
            (component-name (pipe-7))
            (port-name (left-port))
            )
          )
      (port
       (type (liquid))
       (name (out-port))
       (pressure (out-pressure))
       (flow (flow))
       (connection
         (tie-point
           (component-name (pipe-4))
           (port-name (right-port))
           )
         )
       (port
         (type (electrical))
         (name (control))
         (voltage (v-in))
         (connection
           (tie-point
             (component-name (wire-3))
             (port-name (left-end))
```

```
;              )
;            )
;          )
;        );sf-list
;   (parameters
;     )
;   (behaviors
;     (sf-list
;       (behavior
;        (cond ((peq v-in 'high)))
;        (eqns
;          (sf-list
;            (peq in-pressure out-pressure)
;            )
;          )
;        )
;       (behavior
;        (cond ((pneq v-in 'high)))
;        (eqns
;          (sf-list
;            (peq flow 0)
;            )
;          )
;        )
;       );sf-list
;     );behaviors
;   );component
; example of a tank
;(component
   (name (tank13))
;  (type (tank))
   (ports
;    (sf-list
       (port
        (type (liquid))
        (name (in-port))
        (pressure (in-pressure))
        (flow (in-flow))
        (connection
          (tie-point
            (component-name ())
            (port-name ())
            )
          )
       (port
        (type (liquid))
        (name (out-port))
        (pressure (out-pressure))
        (flow (out-flow))
        (connection
          (tie-point
            (component-name ())
            (port-name ())
            )
          )
        );sf-list
```

```
;  (parameters
;    (sf-list
;      (parameter
;       (mass (maximum-mass))
;       (value (1700))
;       )
;  (state-variables
;    (sf-list
;      (state-variable
;       (mass (contents))
;       )
;      )
;    )
;  (behaviors
;    (sf-list
;      (behavior
;       (cond
;         (pand
;           (p< contents maximum-mass)
;           (p< 0 contents)
;           )
;         )
;       (eqns
;         (sf-list
;           (peq (pd/dt contents) (p- inflow out-flow))
;           (peq in-pressure (p* contents .31))
;           (peq out-pressure (p* contents .31))
;           )
;         )
;       )
;      (behavior
;       (cond
;         (peq contents maximum-mass)
;         )
;       (eqns
;         (peq in-flow out-flow)
;         (peq in-pressure out-pressure)
;         )
;       )
;      (behavior
;       (cond
;         (peq contents 0)
;         )
;       (eqns
;         (peq out-flow 0)
;         )
;       )
;      );sf-list
;    );behaviors
;  );component

;syntax of slot-filler objects
; <sf>          ::= ( <header> ( <slot> ( <filler> )) ( <slot> ( <filler> )) ...
; <header>      ::= <symbol>
; <slot>        ::= <symbol>
; <filler>      ::= <atom>
```

```
;                 ::= <sf>
;                 ::= sf-list <sf> <sf> ... <sf>
;                 ::= an expression to be eval-d (this is kludgy)
; <symbol>        ::= denotes a lisp symbol
;
;
;
```

AIRPLANE & ENGINE FUEL SYSTEM

Controls
Wing
CTR
Conf Tank - Norm, Stop trans, stop refuel
Conf Tank emerg trans - Norm, L, R
Ext Trans

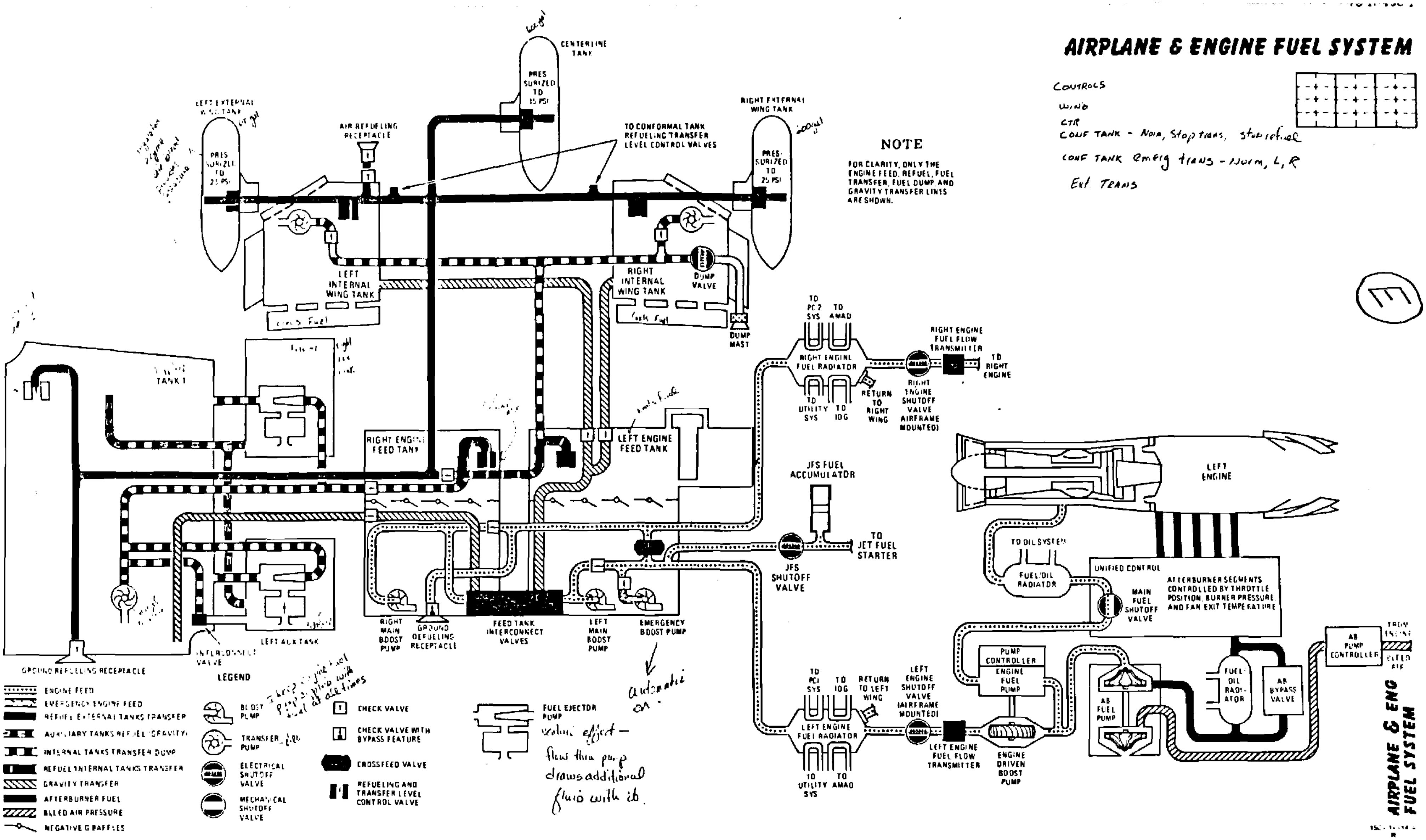


Figure FO-4

F

A DEEP-REASONING AID FOR DEEP-REASONING FAULT DIAGNOSIS

Wan C. Yoon and John M. Hammer
Center for Man-Machine Systems Research, Georgia Institute of Technology, Atlanta, Georgia 30332

ABSTRACT

Wan C. Yoon and John M. Hammer, 1987. A deep reasoning aid for deep-reasoning fault diagnosis. Human-Computer Interaction, Vol 2 (G. Salvendy, ed.)

The design and an experimental evaluation are presented for an intelligent aid for a human operator who must diagnose a novel fault in a physical system. A novel failure is defined as one that the operator has not experienced in either real system operation or training. When the operator must diagnose a novel fault, deep reasoning about the behavior of the system is required. The aid contains features that support such reasoning. One of these is a qualitative, component-level model of the physical system. Both the aid and the human are able to reason causally about the system in a cooperative search for a diagnosis. The human diagnostic performance improved by almost a factor of two when the aid presented the information of observed system behavior or the difference between observed and normal behavior.

1 INTRODUCTION

In highly automated systems, the human operator is primarily a monitor and supervisor [Rasmussen, 1983]. An important monitoring function is diagnosing equipment faults, a difficult task in automated systems. The current approach to fault diagnosis is to train the operator to deal with relatively common faults. The training might teach the operator to use symptoms to distinguish faults and to follow procedures to correct them. While this approach should be successful with common faults, it does not support diagnosis of novel faults. A common sense but unsuccessful approach to help operators diagnose novel fault is to teach them the principles of operation of the system. With this theoretical knowledge, the operators should be able, in principle, to diagnose any failure. Unfortunately, there is little evidence that theoretical knowledge helps operators diagnose failures [Morris and Rouse, 1985a,b]. A logical consequence of this observation might be to put theoretical knowledge into the aid rather than the operator.

Our aid is based on deep, causal reasoning about the system. There are several advantages to this approach. First, novel fault diagnosis is normally considered to be knowledge-based reasoning [Rasmussen, 1983]. Hence, it seems appropriate for an intelligent aid to reason causally. Second, this approach should be more reliable and robust. The system knowledge is represented at the component level. Because components are small and comprehensible, it should be possible to create representations that are correct, perhaps even provably so. These points support the belief that causal reasoning can cover a wider range of faults [Davis, 1984].

In spite of the power of the intelligent aid, we believe there are several reasons to keep the human in command of the problem solving. First, diagnosing a novel failure may require the human to extend the aid's model. Second, when diagnosis involves operating the system (e.g., opening valves, starting motors), it would be better to leave these operations to the human. Third, causal reasoning is slow because the diagnosis problem is a combinatorial search. It may be that the human and the aid may be better able to find a solution cooperatively than either can alone. This is possible, even necessary, for two reasons. The human has better pattern recognition capabilities and can make inductive leaps. Second, the human may need to resolve ambiguities inherent in the aid's model.

In the subsequent sections of this article, we will discuss the system and the experimental task, the interface, the model of human information processing, the aids, and the experimental results.

2 THE SYSTEM AND THE TASK

2.1 The System

The Orbital Refueling System (ORS), a NASA-designed payload on the Space Shuttle, was selected for study [NASA, 1985]. The function of the ORS is to refuel orbiting satellites with hydrazine, with the objective of extending their useful service life. As shown in Figure 1, the ORS fluid system contains a variety of components such as tanks, valves, pipes, etc. The operator controls the simulated ORS by opening and closing valves. Transferring fuel from propellant tank 1 to propellant tank 2 might proceed as follows. First, tank 2 pressure is reduced by momentarily opening valves 10, 11, 13, and 17. Second, tank 1 is pressurized by opening valves 1, 3, and 7. Gaseous nitrogen will flow out of the two small supply tanks, be pressure regulated, and fill tank 1 on one side of the bladder. To transfer fuel to tank 2, valves 5, 14, 15, 16, and 9 would be opened. Because this version of the ORS was for demonstration purposes, all transfers take place between the two large tanks rather than to a satellite fuel tank. There are several assemblies whose purpose was not explained in the above example. The relief valves RV1 and RV2 serve as a safety pressure relief. Check valve CV1 prevents backflow into the gas system. The bladders in tank 1 and 2 serve to isolate the fuel from the propellant and also to contain the fuel in the weightlessness of space. Some components (e.g., valves 10 and 11) may seem redundant; they are so by design for two failure tolerance.

2.2 The Diagnosis Task

The operator's task is to diagnose the failure in the system. This requires the operator to manipulate and observe the system, because a diagnosis cannot be determined uniquely from an observation of a state vector at a single point in time. The diagnosis task is difficult for the following reasons. First, all component testing must be done in the context of the system. It is not possible to remove a component for isolated testing. Thus, every diagnostic test requires nontrivial interpretation. Second, the data are limited and may contain one or more errors. There are seven pressure sensor readings and fourteen commanded valve positions. Both can contain an error. A pressure sensor may report a false reading or a valve may disobey its command. The consequences are that an unaided diagnosis can easily require ten minutes.

3 AIDING WITH A QUALITATIVE MODEL

This section describes the interface, our model of operator information processing, and the aids. The interface has four windows: schematic, interac-

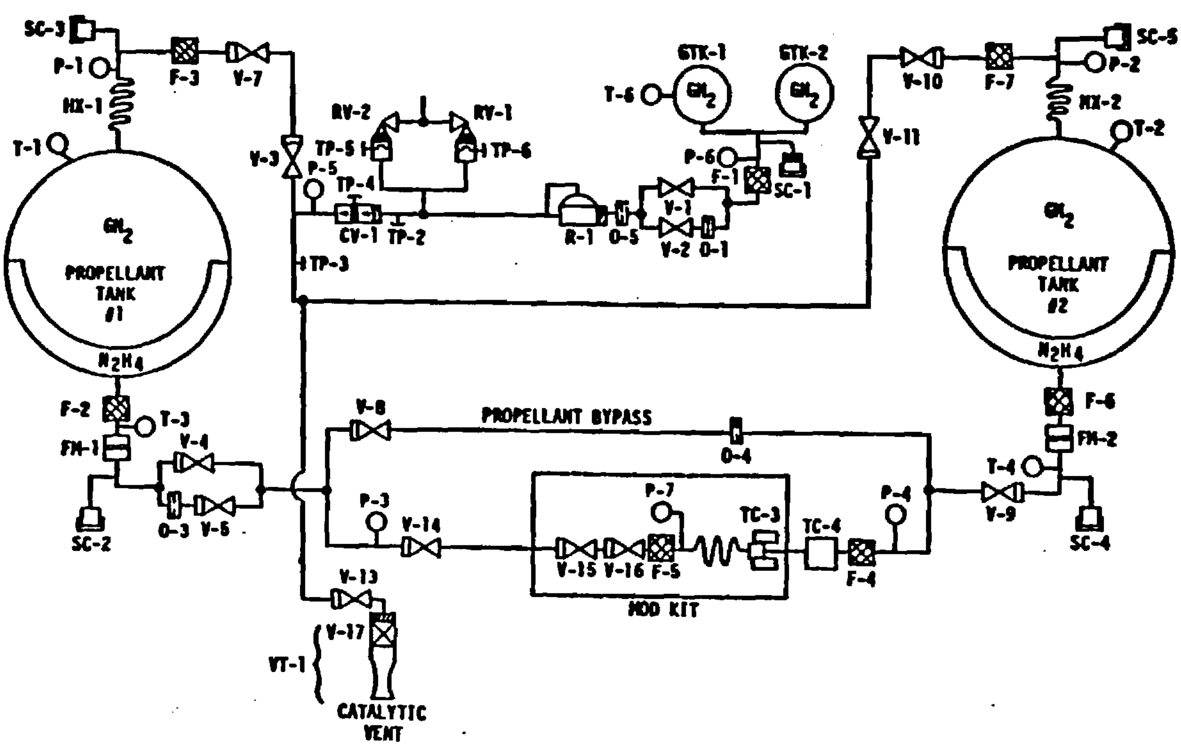


Figure 1. The Orbital Refueling System (ORS).

tion, sensor display, and hypotheses.

The schematic window displays a schematic diagram of the ORS. The schematic always shows the commanded states of the valves. Certain forms of aiding (described below) change the display of paths along which mass may flow. The appearance of the schematic is similar to Figure 1.

The interaction window is where the operator's commands appear. The commands available to the operator include the following:

1. Opening and closing valves.
2. Comparing two pressures. On a real physical system, the numerical pressure could be displayed on the schematic. When a qualitative model is used, there is no scale in general to which a pressure can be referred. Instead, the subject may request the relationship (<, =, >) between two pressures or between a pressure and a nominal reference pressure such as absolute zero or the regulator's design set point.
3. Display of the first derivative of pressure (positive, zero, or negative).
4. Turning the what-if model aid (described below) on and off, and stating hypotheses to the what-if model aid. When the what-if model is on, the open, close, and comparison commands apply both to the system and the what-if model.

The sensor display contains the output from the comparison command: the relationship between two pressures or the first derivative of a pressure. The what-if model, if activated, has its corresponding output displayed side-by-side with the system model.

The hypotheses window will display any hypotheses that the operator expresses through commands in the interaction window.

4 A MODEL OF OPERATOR INFORMATION PROCESSING

4.1 Observation of Strategies

Our model of operator information processing directly influenced the design of the aids. From the observation of diagnostic behavior, we had identified three strategies that subjects used: hypothesis-driven evaluation, data-driven evaluation, and topographic search [Yoon and Hammer, 1987]. Hypothesis-driven evaluation starts with the planning of a test procedure for a given hypothesis. A test plan would be diagnostic if, given that the hypothesis is true, the response of the system to the test is unique to the hypothesis. When a sufficiently diagnostic test has been planned, the test is executed and its result evaluated. Because the hypothesis needs to be explicit enough to enable the prediction of its resulting system behavior, this strategy is mostly used in the later phase of diagnosis.

With data-driven evaluation, the subject first examines a piece of data to determine if it is worth closer attention. This examination is done by comparing the data to the expected system behavior. If the data turns out to be unexpected (i.e., not explained in terms of previously observed symptoms or normal behavior), then hypotheses are formulated to explain the data. Whether the formulation is successful, this piece of data is remembered as another symptom to be used later during diagnosis. Since this strategy does not require a well-formed hypothesis, it was heavily employed in the initial phase of diagnosis.

Topographic search follows the connections between components to track down the source of the malfunction. In contrast to hypothesis-driven and data-driven evaluation, it does not appear to require as deep a reasoning about device behavior. Thus, it is easier.

4.2 Types of information processing

As frequent parts of some of the above strategies, the operator needs presumably to do the following types of information processing. First, the operator must envision the normal behavior (i.e. no failures) of the system. Second, the operator uses external, observable information (i.e., pressure information) to determine unobservable, internal behavior (i.e., presence of a mass flow, a leak somewhere in a path). Third, the operator must form the difference between the observed and normal system behavior. These three forms of processing could be termed N (normal), O (observed), and O-N (observed minus normal).

The aids parallel the above three forms of processing. N and O aiding are intended to help the operator with N and O processing, respectively. Both are displayed in the same way. The schematic display is modified to show both mass flow paths (the movement of either gas or liquid) and equal pressure paths. The determination of these paths is from a system model (N) or pressure observations (O) available to the aid. The aid has exactly the same information as does the operator.

O-N aiding is the difference between observed and normal behavior. This information is displayed in the sensor display window in the form of suggested data observation commands. This form of aiding was also predicted to be useful based on earlier observations [Yoon and Hammer, 1987]. Subjects appeared to have difficulty selecting effective data to observe.

A fourth form of aiding, O-H, is closely related to the third, O-N. O-H (observed minus hypothesized) aiding displays the difference (as described above) between the observed behavior and a system with one or more hypothetical failures. This aid allows the operator to set a hypothesis. If the hypothesis is correct, there will be no difference between observed and hypothesized behavior. This aid gives the operator an unambiguous interpretation of the correctness of a hypothesis. It does not, however, tell the operator how to modify the hypothesis if it is incorrect.

5 EXPERIMENTS AND RESULTS

5.1 Procedure

Two experiments were conducted to evaluate the aids. A comparison of N versus unaided performance was first tested since we had earlier observed that most subjects found it confusing or irrelevant. The more prospective aids, O and O-N, were evaluated in the second experiment. Six and nine engineering students participated in the first and the second experiment, respectively.

Two training sessions preceded the experimental session. The first session was self-paced instruction on basic fluid dynamics and the operation of the ORS. In the second session, the subjects practised testing various hypotheses and solved five diagnostic problems both with and without the aids. The purpose of these experiments intentionally limited the useful range of diagnostic skill of subjects. An overtrained subject tends to develop some mechanistic diagnosis procedures. These may replace the deep reasoning about the system and deal with the problems as routine failures rather than novel ones. With too little training, the subject's performance would reflect more of deficiency in knowledge than the difficulty of the problem solving. For these reasons, the experimenter interacted with the subjects in both training sessions to insure proper understanding of the material.

The subjects started the experimental session with several additional, warm-up exercises and solved six main problems. Keystrokes and verbal protocols were collected. The performance measures were the time to diagnose (TTD) and the number of information gathering actions (#IGA). Problem and subject were blocking variables. Each subject solved the problems with an equal number of different aiding levels. A replicated Latin square was used. Order effects were counterbalanced. Three subjects formed a group, which received the same order of aids, to serve as replications for the evaluation of interaction terms in both experiments [Winer, 1962, pp. 538-543].

5.2 Results

The results of significance tests were same with TTD and #IGA. The effect of N aiding was somewhat negative, though not significant. Most subjects said after their sessions that the aid N was rather confusing or that it was not the information they were seeking during the diagnosis. Both O and O-N aids showed a positive improvement in diagnostic performance at the 0.05 significance level. The effects of both blocking variables, subject and problem, were significant. But, there was no significant interaction between any two variables. Residual analysis revealed that logarithmically transformed data better satisfied the homogeneous variance assumption. No test results, however, were changed by the transformation. It was shown that O-N and O shortened TTD on the average by 42% and 34%, respectively. The aiding effects appeared similar in #IGA: 44% decrease with O-N, 40% with O.

5.3 Additional observations

The following observations, while not the result of hypothesis testing, were also made during the course of the experiment. The aids more benefited the problem solving earlier in the diagnosis. This was expected because one of the effects of O and O-N was to reveal abnormal system responses, and thus to stimulate the subject to launch a data-driven evaluation. In fact, O-N aiding obviously encourages the subject to select meaningful data,. Toward the end of diagnosis, the subjects developed explicit hypotheses (i.e. hypothesis-driven evaluation), and tended to be too heavily involved in their own testing procedure to pay attention to the aid. In fact, the aiding information is usually no longer relevant to the subjects' highly detailed hypothesis testing. To aid the final phase of diagnosis, the aid needs to know the operator's hypothesis. Then, the aid could run a modified qualitative model according to the hypothesis (H) and calculate its deviation from the observed behavior (O). The difference O-H may be more relevant than O-N to the later phase of diagnosis.

6 CONCLUSION

An aiding approach has been described and evaluated for novel fault diagnosis in complex systems. To the best of our knowledge, this approach is unique in the following ways. First, the emphasis is on novel rather than routine faults. Second, it contains a qualitative model that may correspond to the human's internal model of the system. This model represents knowledge only of how the system behaves. Therefore, this aiding approach does not rely on proceduralized knowledge. Third, the qualitative model is the basis for much of the aiding that takes place.

The qualitative model was used to help different tasks of human information processing. Presentation of observed system behavior (O) improved the diagnostic performance of subjects, while that of normal system behavior (N) does not. One implication is that the prediction of current actual system behavior is a task that needs more help. Aiding of envisioning normal system behavior according to commanded physical configuration is less effective and, when emphasized saliently, seems to interfere with the diagnostic reasoning. Pointing out the abnormality in the observed system behavior (O-N) was at least as effective as O.

More generally, the experiment confirmed that a deep reasoning diagnosis can be aided, without disturbing the human diagnostic procedure, by providing relevant information. It should be emphasized that this was possible through an understanding of the operator's information needs and that a qualitative model could be used to generate the information that seemed to be well accepted for augmenting the human's mental model.

7 ACKNOWLEDGMENT

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

8 REFERENCES

Davis, R., 1984. Diagnostic reasoning based on structure and behavior. Art. Intel., 24: 347-410.

Morris, N.M. and Rouse, W.B., 1985. The effects of type of knowledge upon human problem solving in a process control task. IEEE Trans. Sys., Man. and Cyber., 15(6): 698-707.

Morris, N.M. and Rouse, W.B., 1985. Review and evaluation of empirical research in troubleshooting. Human Factors, 27(5): 503-530.

NASA Johnson Space Center, 1985. Orbital refueling demonstration system description. Program Development Office.

Rasmussen, J., 1983. Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models. IEEE Trans. Sys., Man. and Cyber., 13-3: 257-266.

Winer, B.J., 1962. Statistical Principles in Experimental Design. McGraw-Hill, New York/San Francisco/Toronto/London.

Yoon, W.C. and Hammer, J.M., 1987. Using a Qualitative Model to Aid the Operator. to appear in the IEEE Trans. Sys., Man. and Cyber. during Novel Fault Diagnosis. Proceedings of the International Conference on Cybernetics and Society, pp. 1222-1227.

E-2

NASA Grant NAG 2-123*

PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

FINAL REPORT

John M. Hammer, Principal Investigator

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

*The NASA Technical Officer for this grant is Everett Palmer, NASA Ames Research Center, MS 239-2, Moffett Field, CA 94035.

Introduction

This research investigated ways in which computers can aid the decision making of an human operator of an aerospace system. The approach taken is to aid rather than replace the human operator, because operational experience has shown that humans can enhance the effectiveness of systems. As systems become more automated, the role of the operator has shifted to that of a manager and problem solver. This shift has created the research area of how to aid the human in this role.

The remainder of this report describes published research in four areas. It concludes with a discussion of the DC-8 flight simulator at Georgia Tech.

Archival Literature

1. Lewis, C.M., and Hammer, J.M. (1986). Significance testing of rules in rule-based models of human problem solving. IEEE Transactions on Systems, Man, and Cybernetics, SMC-16(1).

2. Yoon, W.C., and Hammer, J.M. (1988). Aiding the operator during novel fault diagnosis. IEEE Transactions on Systems, Man, and Cybernetics, SMC-18(1).

3. Yoon, W.C., and Hammer, J.M. (1987). A deep reasoning aid for deep-reasoning fault diagnosis. In G. Salvendy (Ed.), Human-Computer Interaction, Vol. 2, Amsterdam: Elsevier.

4. Yoon, W.C., and Hammer, J.M. (1988). Deep-reasoning fault diagnosis: An aid and a model. To appear in IEEE Transactions on Systems, Man, and Cybernetics.

Conference Literature and Technical Reports

5. Knaeuper, A., and Morris, N.M. (1984). A model-based approach for online aiding and training in process control. Proceedings of the 1984 International Conference on Systems, Man, and Cybernetics. Halifax, Nova Scotia, Canada.

6. Lewis, C.M. (1985). Rule-based analysis of pilot decisions. Proceedings of the Human Factors Society 29th Annual Meeting.

7. Lewis, C.M. (1985). Issues in rule identification and logical induction. Proceedings of the 1985 IEEE International Conference on Systems, Man, and Cybernetics.

8. Lewis, C.M. (1986). Identification of rule-based models. Unpublished Ph.D. dissertation. Atlanta, GA: Georgia Institute of Technology.

9. Rouse, W.B., and Morris, N.M. (1985). Understanding and avoiding potential problems in implementing automation. Proceedings of the 1985 International Conference on Systems, Man, and Cybernetics.

10. Yoon, W.C., and Hammer, J.M. (1985). Aiding the operator during novel fault diagnosis. Proceedings of the 1985 International Conference on Systems, Man, and Cybernetics.

11. Yoon, W.C. (1987). Aiding the Operator During Novel Fault Diagnosis. Unpublished Ph.D. dissertation. Altanta, GA: Georgia Institute of Technology.

Model-Based Online Aiding [5]

This research addressed the feasibility of adapting an existing rule-based system as an online "coach" for controlling PLANT, a simulation of a generic process plant. KARL, a rule-based model capable of controlling PLANT, was adapted to provide three types of information to subjects:

1) situation assessment (i.e., which operational procedure, if any, was applicable for a given situation);
2) guidance in following procedures (i.e., feedback whenever subjects' actions were inconsistent with available procedures); and
3) performance feedback (based upon changes in the system's stability).

Subjects received this information online while controlling PLANT. Compared to subjects in an earlier experiment who controlled PLANT without the benefit of the coach, these subjects maintained a generally more stable system, scored higher on a paper-and-pencil test of system knowledge, and were more successful in diagnosing an unfamiliar failure of the PLANT safety system. Careful analysis of these results in light of previous research with PLANT indicated that the reasons for these differences were not as straightforward as they might appear. This experiment is viewed as illustrating potential benefits and subtleties of using a rule-based model as an online coach.

Significance Testing of Rule-Based Models [1]

Many researchers have used rule-based systems to model human problem solving. Typically, the rule-based system has a large number of rules, each of which has several free variables that were adjusted during the modeling process. For the most part, significance testing of these rules has not been much of a consideration, although it should be. It is possible to describe N data perfectly with N rules using a trivial model that simply reproduces the data. While there is no evidence that this has happened in any of the research reported to date, there is a certain danger of overfitting a rule-based model.

Three methods were developed for testing the statistical significance of rules and other components of rule-based models. It was assumed that the percentage of behavior matched (e.g., commands) was the performance measure of interest. Two of the testing approaches, however, were not limited to this measure. They may be used to study any performance measure, though it may be possible for a rule to produce a statistically significant effect on one performance measure but not another. Rule testing by analysis of variance, randomization, and contingency tables was studied, and comparisons between these methods were developed.

Identification of Rule-Based Models of Problem Solving [6, 7]

Rule-based models have frequently been used to model human performance and behavior. A machine learning program was used to identify the rules employed by humans in two settings. The first setting was a collision avoidance maneuver for which the pilots had a cockpit display of traffic information (CDTI). This data was generated from an experiment to evaluate the effects of various CDTI displays on avoidance behavior.

The rules produced by the machine learning program can be combined in a decision sequence that accounts for a substantial portion of the maneuvers. When the intruder was maintaining a constant altitude, pilots executed vertical away maneuvers even for intruders posing no threat. This is the easiest of the maneuver decisions because it entails no geometric complications and was used whenever possible. For intruders changing altitude, a minority of pilots consistently checked for a threatening separation and remained on course if none existed. Another subgroup responded to horizontal threats by uniformly turning toward the intruder. This is a good decision if the intruder would have passed in front but aggravates the situation for intruders which would pass behind. The remainder of the pilots included this qualification in their decisions to turn toward the intruder. The mirror of this strategy, turning away from intruders which would pass behind was not observed.

The second setting was PLANT [Morris, N.M., and Rouse, W.B. (1985). "The effects of type of knowledge upon human problem solving in a process control task." IEEE Transactions on Systems, Man, and Cybernetics, SMC-15(6).], a simulated industrial process in which feedstock is pumped in at one end and the finished product is pumped out at the other. A three-by-three matrix of tanks connects PLANT input to output. Each tank is connected by valves to all tanks in adjacent columns. The operator controls valve positions and pumping rates for feedstock and product. Fluid dynamics are modeled within the system causing lags and oscillations to result when valves change state, as well as varying rates of flow due to relative tank heights. Valves trip closed when flow exceeds their setpoints. Failures of pumps and valves are also possible. The CRT system display shows tanks, their levels of fluid, open valves connecting the tanks, and numerical labels showing pumping rates and tank levels.

In concert these features produce a complex symbolic task in which conflicting goals relating production, system stability, long term trends, failures, and trips must be balanced to operate the system. At peak efficiency, all valves should be open, tank levels uniform across the system, and identically high pumping rates set for feedstock and product. PLANT is operated by subjects through a services of iterations which a control action is entered and the resultant updated system state displayed. The iterations from an experimental session (~500) provide a series of "snapshots" isolating specific system states and the responses subjects made to them.

In an initial analysis of this data [8], small sets of high coverage rules were assembled. Cross-validation was used to assess the reliability of the selected rules. Identified rules correctly matched 51% of control decisions in the identification sample for subjects in the control group and 32% of the control decisions in the validation sample. For subjects using PLANT procedures, combining symbolic (rule-based) and signal (internal dynamic model of PLANT) processing fared better matching control decisions 52% of the time. The generality of the well-performing rules obtained prohibited the detailed analysis of strategy possible in the CDTI case.

Deep Reasoning Fault Diagnosis [2, 3, 4, 9, 11]

This research studied the design and evaluation of knowledge-based aiding for a human operator who must diagnose a novel fault in a dynamic, physical system. Since the operator must employ deep reasoning about system behavior to diagnose such a fault, his or her performance may be restricted by cognitive limitations and biases. A computer aid based on a qualitative model of the system was built to help the operator overcome some of these limitations. This aid differs from most expert systems in that it operates at several levels of interaction which are believed to be more suitable for deep reasoning.

Four aiding approaches, each of which provided unique information to the operator, were evaluated. The aiding features were designed to help the human's causal reasoning about the system in predicting normal system behavior (N aiding), integrating observations into actual system behavior (O aiding), finding discrepancies between th two (O-N aiding), or finding discrepancies between observed behavior and hypothetical behavior (O-H aiding). Three experiments were conducted to evaluate the aiding approaches and to investigate the nature of deep-reasoning diagnosis. Human diagnostic performance improved by almost a factor of two with O aiding and O-N aiding. The results from the experiments were integrated into a model of human information processing in causal reasoning diagnosis.

DC-8 Flight Simulator

The failure to both complete and utilize the DC-8 flight simulator is a disappointment. An assessment of the cost of developing the simulation should have been prepared initially. The development breaks down into three categories: hardware, flight simulation, and display generation. The hardware category was completed at a cost of roughly $75,000. The flight simulation code is roughly one half done, and perhaps another 10,000 lines of code need to be written and tested. This would require one programmer-year to produce ($50,000). Display generation would require $15,000 in hardware and another programmer-year ($50,000). A total estimated cost of $190,000 compares favorably with the cost of a commercial product. However, the research funding needed to support such a facility must be larger than a single $100,000/year grant.

DEEP-REASONING FAULT DIAGNOSIS: AN AID AND A MODEL

Wan C. Yoon and John M. Hammer

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

ABSTRACT

The design and evaluation are presented for knowledge-based aiding for a human operator who must diagnose a novel fault in a dynamic, physical system. Since the operator must employ deep reasoning about system behavior to diagnose such a fault, the performance may be restricted by cognitive limitations and biases. A computer aid based on a qualitative model of the system was built to help the operator overcome some of his/her cognitive limitations. This aid differs from most expert systems in that it operates at several levels of interaction which are believed to be more suitable for deep reasoning.

Four aiding approaches, each of which provided unique information to the operator, were evaluated. The aiding features were designed to help the human's causal reasoning about the system in predicting normal system behavior (N aiding), integrating observations into actual system behavior (O aiding), finding discrepancies between the two (O-N aiding), or finding discrepancies between observed behavior and hypothetical behavior (O-H aiding). Three experiments were conducted to evaluate the aiding approaches and to investigate the nature of deep-reasoning diagnosis. Human diagnostic performance improved by almost a factor of two with O aiding and O-N aiding. The results

from the experiments were integrated into a model of human information processing in causal reasoning diagnosis.

INTRODUCTION

Becoming more of a monitor and supervisor in today's highly automated systems [Rasmussen 1984], the human operator must at times be involved in the task of diagnosing system failures, which is increasingly difficult as the system becomes more complicated and automated. The prevalent approach to fault diagnosis is to train the operator to have better knowledge and experience with commonly expected faults. The training might teach the operator to use symptoms to distinguish faults and to follow procedures to correct them. While this approach should be successful with common faults, it does not support diagnosis of novel faults.

Another, more recent approach is to support the human operator via expert systems for diagnosis. Those expert systems are typically based on a large collection of diagnostic rules, which associate symptoms to causes and generate tests. As for novel failures, many expert systems for diagnosis [Shortliffe 1976, Miller, Pople, and Myers 1984] are based on shallow reasoning: a set of symptoms suggests a diagnosis. This mapping is based on experience rather than a system model. Consequently, such systems are subject to the same limitations as training and procedures. The expert system designer has to anticipate the failure for the expert system to solve it correctly.

Aiding Based on a System Model

To diagnose an unanticipated, unexperienced fault, the operator must rely on his/her understanding of causality of the system [Davis 1984]. Such causal reasoning is usually a very demanding cognitive task when the system is complex. Therefore, an intelligent aid should be able to support the operator in causal reasoning about the system behavior. The most obvious way to achieve this is to let the aid run its own causal model of the system and provide the results to the human. A qualitative model of the system can be useful for this purpose.

Another advantage of an aid based on a causal model is that it should be more reliable and robust. The system knowledge is represented at the component level. Because components are small and comprehendable, it should be possible to create representations that are correct, perhaps even provably so. A system fault can be expressed as a combination of component faults which does not require a priori identification of the system fault itself. Thus, an aid based on a causal system model can cover a wider range of faults.

In spite of the power of the intelligent aid, we believe there are several reasons to keep the human in command of the problem solving. First, the current trend of automatic diagnosis is based on large rule-bases which are less useful in novel fault diagnosis. Second, the human and the aid may be better able to find a solution cooperatively than either can alone. This is possible, even necessary, because the human has better pattern recognition capabilities and can make inductive leaps. Third, in many cases, diagnosis is one of the subgoals and may interfere with other subgoals. For example, when diagnosis involves operating the system (e.g., opening valves, starting motors), it may interfere with the subgoal of system safety. The human is better suited for the responsibility of resolving tradeoffs in pursuit of an overall goal. Lastly, the human may need to resolve ambiguities inherent in the aid's model or even to extend the model.

Suboptimalities in Human Problem Solving

The aid is designed to mitigate human suboptimalities that occur during decision-making and troubleshooting [Wickens 1984]. Two categories of suboptimalities used here are knowledge-limited and cognition-limited. The knowledge-limited suboptimality is simply that the operator does not fully understand the system. Obviously, the aid's model is a basis for compensating for this problem.

Cognition-limited suboptimalities are of more interest when the system fault is novel rather than common. Novel fault diagnosis requires causal reasoning about the system, which is a cognitively very demanding task. The operator should repeatedly run a mental model of the system in multiple modes as well as maintain a diagnostic procedure. The required information processing can overload the operator's limited mental resources, especially attention and working memory. The results may be incorrect reasoning or inefficient use of information.

To help, the computer aid can process and display useful information so that the operator can use it. This may improve the system performance in two ways. First, the operator can dynamically allocate some subtasks to the aid and concentrate on others. This leads to lessened mental workload and improved performance on those subtasks undertaken by the operator. Second, since the aid reasons in parallel with the human, the human can confirm his/her results against the aid's results. When the human overlooks some useful information or is affected by some biases, discrepancies would be noticed between the aid's results and the operator's own. The operator may then adopt the aid's result to be used in subsequent reasoning. For example, when the human and the aid evaluate a hypothesis, the confirmation bias (i.e., the tendency to seek only confirming evidences) will be prevented since the aid, being not susceptible to this bias, would report disconfirming evidence.

Research Questions

It is likely that not every plausible form of aiding will improve operator performance. Some information which is both relevant and helpful may not be able to improve human performance because the human fails to incorporate the information into his/her problem solving. This leads to another question: which types of information are easily usable by the human? Our approach to answering these questions was, first, to build an aid based on the best principles available to us, and let the aid supply prospective types of information in experimental settings to evaluate their actual aiding effects. Successful and unsuccessful aiding may also provide insight on the architecture of human information processing.

In the subsequent sections of this article, we will discuss the suitable form of interaction for a deep-reasoning aid, the system which served as the context of problem, qualitative modeling of the system, the features of the aid, the experiments and results, and a model of human information processing in causal reasoning diagnosis. Because a literature review was included in recently published, early report of this research [Yoon and Hammer 1987], no review appears here.

LEVELS OF INTERACTION

In the design of interaction between the aid and the human, it is important to consider the nature of task to be aided. Deep-reasoning diagnosis has many subprocesses of which even the problem solver may not be aware. The aid should be able to help the human's processing without disturbing or interfering with it.

To discuss appropriate forms of interaction in this situation, we stratify the ways in which the human and computer interact into five levels in terms of intrusiveness (Figure 1). The two extreme (i.e., the most intrusive) levels are the human-direct level and the computer-direct level. In the middle, the human-suggest and the computer-suggest levels allow a problem solver, the human or the aid, to be moderately intrusive. Finally, there is the independent level at which neither problem solver influences the other. This stratification is orthogonal to the levels of required intelligence or knowledge the aid should have [Greenstein 1980].

At the <u>human-direct level</u>, the human assigns tasks to the computer. For example, the computer will respond to the operator's request to perform a subtask or to answer a question. The situation is opposite at the <u>computer-direct level</u>; the computer asks the human for some information or to perform some tasks. The human does not have a choice other than to follow the request.

Typical expert systems use only these two levels of interaction; some systems use only one of the two, others use both. At either level, the overall processing is serial and requires explicit communication. Certainly, this property does not promote the human's deep reasoning. The difficulty of human-direct level interaction is that the effectiveness of the aid depends upon the ability of the human to decompose the overall task into modular subtasks [Wickens 1984]. On the other hand, at the computer-direct level, the human does not have the freedom to pursue his/her own processing. This would reduce the benefit to the system of having the human whose flexibility and inductive and pattern recognition capabilities are superior to those of automation.

At the human-suggest level of interaction, the human may impose constraints on the computer's processing. Examples are adjusting weights of different criteria, modifying the computer's intermediate results, or restricting the computer's attention to some area in the problem space. However, the computer will continue its tasks without explicit assignment by the human; only the data or criteria are modified. The computer-suggest level allows the computer to provide some information or warning to the human. The human is free to attend or not depending on his/her assessment of situation. The operator may postpone a response until finishing a current line of reasoning; or, the computer can be completely ignored. Thus, the communication is allowed to be less explicit and more abstract. What becomes a critical issue is that the suggestions by the computer need to be compatible with the human's reasoning process.

At the independent level, both problem solvers pursue their own problem solving procedures without influencing each other. This level is almost non-existent in conventional expert systems which employ only the two extreme levels. When the interaction occurs at the suggest levels, however, the independent level fills the intermissions between suggestions. While there is no interaction, both problem solvers may be highly active in their problem solving. At times, the deep-reasoning process needs to be supported by interruption-free independence.

We believe that the three middle levels should facilitate more adequate aiding to deep-reasoning tasks. At those levels, the processing is more parallel and both problem solvers have more freedom. Two human problem solvers would interact mostly at those levels; they would suggest, take comments and hints, or be silent. Using the three levels of interaction was one of our principles in building the aid for novel fault diagnosis. Another related principle was to consider compatibility of aiding information with human information processing.

THE SYSTEM AND THE TASK

The Orbital Refueling System (ORS), a NASA-designed payload on the Space Shuttle, was selected for study [NASA 1985]. The function of the ORS is to refuel orbiting satellites with hydrazine, with the objective of extending their useful service life. As shown in Figure 2, the ORS fluid system contains a variety of components such as tanks, valves, pipes, etc. The operator controls the simulated ORS by opening and closing valves. Transferring fuel from propellant tank 1 to propellant tank 2 might proceed as follows. First, tank 2 pressure is reduced by momentarily opening valves 10, 11, 13, and 17. Second, tank 1 is pressurized by opening valves 1, 3, and 7. Gaseous nitrogen will flow out of the two small supply tanks, be pressure regulated, and fill tank 1 on one side of the bladder. To transfer fuel to tank 2, valves 5, 14, 15, 16, and 9 would be opened. Because this version of the ORS was for demonstration purposes, all transfers take place between the two large tanks rather than to a satellite fuel tank. There are several assemblies whose purpose was not explained in the above example. The relief valves RV1 and RV2 serve as a safety pressure relief. Check valve CV1 prevents backflow into the gas system. The bladders in tank 1 and 2 serve to isolate the fuel from the propellant and also to contain the fuel in the weightlessness of space. Some components (e.g., valves 10 and 11) may seem redundant; they are so by design for two failure tolerance.

Nomenclature

In discussing the ORS and the operator's actions and diagnosis, we have found the following nomenclature useful. A component is the smallest unit of the ORS system that is modeled in isolation. Typical components include valves, tanks, pipes, regulators, sensors, etc. The entire set of components, working together according to the qualitative dynamics, is a system. A path is a connected set of components, which could be either a graph-theoretic path or tree.

Components have states. For example, a valve may be open, closed, or leaking. The state is what the component is actually doing. A commanded state is the state to which a commandable component asked to assume. For example, a valve may be commanded open or closed. A component also has a behavior mode, such as fail-open or normal. The behavior mode describes the states which the component takes in response to commands and external conditions. For example, a fail-open valve is always open, regardless of the command.

The Diagnosis Task

The operator's task is to diagnose the failure in the system. This requires the operator to manipulate and observe the system, because a diagnosis cannot be determined uniquely from an observation of a state vector at a single point in time. A solution is an assignment of states to components such that the assignment's behavior is always identical to system behavior. For a single valve failure, the solution would be a normal state for all components save the failed valve, which might be jammed shut. The diagnosis problem can be viewed as a combinatorial search for a state assignment. The search is constrained by the laws of component physics. That is, a state assignment to a component imposes constraints on its neighboring components. For example, if a valve is opened and permits a flow down a pipe, the component receiving the flow must be in a state to accept the flow.

QUALITATIVE MODELS OF CONTINUOUS PHYSICAL PROCESSES

This section describes qualitative models: representations, the computational problems solved, and the specific needs of our aid of the qualitative model.

A qualitative model is a symbolic representation of a system. Its most basic description is of a component. A component is described in terms of its connections to other components and its behavior. Behavior is described in terms of the physical variables which are present at its connections. The differentiation between the structural description (connections) and the behavioral description is particularly important for insuring the robustness of a qualitative model. The isolation of each component in the behavioral description has usually been emphasized by other qualitative modeling [De Kleer and Brown 1983]. Contrarily, our qualitative model represents the system at both the component level and at an aggregated level as paths. The motivation for this is the belief that a multi-level description is closer to the operator's internal model of the process. In fact, more effective communication between our model and the human operator was enabled by the use of the higher level description.

From a given state, the behavior of a component is described in terms of the physical variables present at its ports. A physical variable (and its time derivative) may take several values. The time derivative usually has only one of three possible values: negative, zero, or positive. The variable itself may take either nominal or ordinal values. The nominal values usually correspond to points at which behavior (component or material) changes. For example, water temperature would have nominal values at freezing and boiling. Variables may also take on ordinal values (or relationships). For example, water temperature could be taken to be greater than freezing and less than boiling.

The nominal and ordinal values taken by physical variables are said to occur in a quantity space [Forbus 1984, Kuipers 1984]. The quantity space is a partial ordering on the physical variable values it contains. The partial ordering occurs because not all comparisons are relevant to understanding the physical system qualitatively. For example, consider a valve between two tanks, A and B. When the valve is opened, the resulting behavior is determined by the pressures in two tanks. The pressure at other unconnected points in the system is unrelated to the above behavior.

AIDING WITH A QUALITATIVE MODEL

This section describes how a qualitative model is used as a foundation for aiding. First, each window of the interface will be described. Four different aiding strategies and the motivation for each of them will then be presented. Each strategy emphasizes different type of aiding information.

ORS Interface

The interface has four windows: schematic, interaction, sensor display, and hypotheses (Figure 3). The schematic window displays a schematic diagram of the ORS. The schematic always shows the commanded state of the valves. The interaction window is where the operator's commands are echoed by the interface. The commands available to the operator include the following:

(1) Opening and closing valves.

(2) Comparing two pressures. On a real physical system, the numerical pressure could be displayed on the schematic. When a qualitative model is used to simulate the physical system, there is no absolute scale in general to which a pressure can be referred. Instead, a pressure can be compared to other pressures in the system by the relations less-than, equal-to, or greater-than.

(3) Display of the first derivative of a pressure (positive, zero, or negative).

And, when the corresponding aiding feature (it is described more fully in a later section) is available,

(4) Turning the what-if model on and off.

(5) Making state assumptions in the what-if model.

The sensor display contains the output from the sensor display commands: the relationship between two pressures or the first derivative of a pressure. When appropriate aiding features are activated, suggested sensor readings will also be displayed in this window.

'e hypotheses window displays a set of hypotheses that are set by the operator. These hypotheses are simply state assignments to components (e.g., valve 13: leaking). Pipes, which do not have names displayed in the schematic, are designated as left or right to named components such as valves and orifices. For example, the pipe between valve 8 and orifice 4 is designated either R V8 or L O4.

Aiding Approaches

Based on observed human strategies of diagnosis, four aiding approaches seemed to deserve evaluation. Each approach emphasizes different information and uses an appropriate communication mode for the kind of information.

Topographic Aiding. The first and second aiding approaches are based on two presumed forms of operator cognitive processing. First, the operator must observe and infer what the system is actually doing. This processing is termed O (Observed) and is concerned with flows, leaks through valves, leaks out of pipes, and the general vicinity of the fault. Second, the operator needs to generate normal system behavior to compare with observed behavior. This processing is termed N (Normal). Two obvious forms of aiding are to generate O and N so that the operator does not have to devote cognitive processing to generating them. To produce O, the aid integrates the information from the pressure sensors to which it has continuous access. Like a human operator, the aid has to guess the actual behavior from the sensor information since it does not know the real system state. In contrast, N is generated by the qualitative model under the assumption that every component is in the normal behavior mode.

O and N are displayed topographically. For both O and N, the aid displays two forms of system behavior: equal pressure paths and mass flow paths. The former is the set of components that should be at equal pressure given the commanded valve positions. Whenever the operator creates an equal pressure path by opening a valve, the path is highlighted. Similarly, a mass flow path created by an operation is highlighted as long as it exists.

Figure 4 is an example of N display. Opening valve 9 was the latest change. This would make, if the system were fault-free, the pressure is equal through the highlighted path.

Figure 5 shows the same configuration as Figure 4, except that the O display (rather than N) is activated. Suppose that when valve 9 was opened, the pressure P2 began to decrease and P1 increase. This leads the aid to believe there is a mass flow from tank 2 to tank 1 (the path is highlighted) in spite of the closed positions of valve 8 and valve 15. However, since the aid cannot be certain which valve is leaking, it highlights both paths. When a precise conjecture is not possible, the aid will take a conservative position as in this example. Note that O and N aiding cannot be used simultaneously.

<u>Differencing Observed and Normal Behavior</u>. The third aiding approach is to suggest observations that reveal the differences between the observed system behavior and the normal system behavior. This difference will be referred to as O-N. The importance of O-N in ORS diagnosis was discussed in connection with the results of our preliminary experiment [Yoon and Hammer 1987]. Such a deviation from normal behavior, when observed and correctly interpreted, helped effectively reduce the size of the feasible hypothesis

set. Figure 6 shows an example of this feature's display in the same situation as of Figure 4 and 5. The aid suggests, for example, to issue a command D P1, which is to inquire the first derivative of P1. When the operator follows this, he/she will find P1 is increasing, which is opposite to the commanded situation (no flow should be possible from either GTK or TK2G/L).

The What-if Model. The fourth, and the last, aiding feature is closely related to the above. This feature can use any hypothetical behavior (denoted by H), instead of the normal behavior, with which to difference the observed system behavior. The operator can freely set or remove hypotheses. Then, the aid will run a what-if model based on the hypotheses in place of the normal model. Any discrepancies (denoted by O-H) will be reported in the same way. If the hypothesis is incorrect and the observed and hypothesized bevavior differ, the aid will recommend readings that indicate the difference. If the hypothesis is correct, the aid will produce no recommendations. For example, suppose valve 8 is leaking to allow a flow from tank 2 to tank 1. If the operator's hypothesis is a leak in the pipe between valve 10 and 11, the feature would present a display shown in Figure 7. If the hypothesis were right, P1 should not increase. In this example, P1 does increase, so the aid recommends a reading D P1. Also, the hypothesis does not explain the difference between P2 and P4. Note that if no hypothesis is stated, the recommendations would be the same as the previous example (i.e., O-H = O-N if H = N).

The common motivation for these aiding approaches is to perform computations that the operator is believed to do when diagnosing the system. As much as these computations are related to the human's mental model, the qualitative model in the aid may be an appropriate vehicle to help or replace the computations. There are two ways this approach might help. First, the operator may have an incorrect or incomplete mental model. Second, the operator may have difficulty integrating correct component behavior into correct system behavior because of cognitive limitations. The aiding approaches support different uses of the mental model: to envision the normal or hypothetical behavior, to conjecture the actual behavior, and to describe the difference between behaviors of two (e.g., O and H) models. This does not mean the operator need not understand the system at all; he or she still needs to understand the meaning of aid's information and select the hypotheses.

THE EXPERIMENTS

Overview of Experimental Design

To evaluate the types of aiding information, three separate experiments were conducted. The first experiment tested the effects of N information. The next experiment compared the effects of O and O-N against unaided diagnosis. The last experiment focused on hypothesis testing and evaluated the aiding effects of O-N and O-H.

The display of aiding information prevented those features from being tested together. A subject must not be exposed to both N and O features since severe interference, perhaps in the form of a carry-over effect, was expected. This is because the display of O and N information is identical but each carries a different meaning. O-H and O-N for the same reason should not be used together. When O-H is used, it acts as O-N until the subject expresses one or more hypotheses. This makes a direct comparison between O-N and O-H difficult. Even if O-H really improves the performance, its contribution will be depend on the extent to which a subject uses it. Therefore, a different experimental setting needs to be employed to evaluate the potential benefit of O-H. The above considerations led to the three separate experiments.

In all three experiments, replicated Latin square designs were employed [Edwards 1972]. Differences in the complexity of problems and differences between users were expected to introduce large variation in the performance. It was therefore desirable, in order to enhance the efficiency of the experiments, to select problem and subject as two blocking variables. Such designs are called within-subjects designs for each subject serves in more than one treatment level.

A Latin square design, if its assumptions hold, should be more economical than a corresponding complete block design. Even without considering economy, our experiment does not allow a complete block design. Because a subject should not be given a same problem more than once, he/she can be assigned only one level of treatment for each problem.

In a Latin square design, the positions of each treatment level are counterbalanced: namely, each treatment occurs at each test position with equal frequency. This prevents possible practice effects from being confounded with treatment effects. Instead, practice effects are then confounded with test positions (i.e., problem). However, the problem factor is merely a blocking variable and we were not interested in the significance of its effects. Also, the training was designed to stabilize the subject's performance and thus minimize learning effects.

One possible problem with a within-subject design is that the value of an observation for one treatment may be influenced by the effects of treatments applied during earlier periods. When this arises, the treatment is referred to as having carry-over effects. The influence of this effect, if any, may be partially compensated for by adopting a balanced Latin square design, in which each treatment follows every other treatment the same number of times. When the number of treatments is odd, then at least two Latin squares are required to achieve this. This replication also permits a larger number of data points. All our experiments were designed following the above principles. The resulting designs are presented in Figures 8, 9 and 10.

While the balanced Latin square designs may compensate for the above problems, they are based on several assumptions. A key question concerning the Latin square design model is whether the effects of blocking variables and treatments are additive: since there is only one observation per cell, a Latin square design model assumes additivity to estimate the error variance. If nonadditivity is present in the data, the use of a model assuming additivity will lower both the significance level and the power of the test for treatment effects. Thus, the Tukey test for additivity was conducted whenever we applied a model to the data [Neter and Wasserman 1974, pp.780].

While homogeneity and normality of error variance are the basic assumptions in an ANOVA model, it is known that the F test is not much affected by deviation from these conditions [Lee 1975, pp.284]. However, a residual plot of error terms against expected cell means can reveal the need for transformation of dependent variables. Since a transformation would affect the interpretation of treatment effects, residual plots were examined in every analysis.

Experiment 1

The purpose of this experiment was to compare N aided and unaided diagnosis. It is reasonable to expect diagnostic performance to be improved when the envisionment of normal system behavior is improved. In our pre-experimental observations, however, we observed that most subjects found this aiding confusing or irrelevant. Since its effectiveness was doubtful based on this observation, it was evaluated first.

Six industrial engineering students volunteered to serve as subjects. They were trained through two sessions (total 3.5 ~ 4.0 hours) to acquire enough knowledge of fluid dynamics and elements of diagnostic procedure. The goal of our training was to teach the subjects correct causal reasoning about the ORS and give them reasonably stabilized diagnostic skills. However, if a subject is exposed to a kind of problem several times in a short period, the subject may develop some mechanistic diagnosis procedures that do not require causal reasoning. When a similar problem is given, the subjects may try to deal with it as a routine failure rather than a novel one. We felt that a longer training may increase this possibility since the complexity of our version of ORS is only moderate.

Training session 1 started with basic principles derived from fluid dynamics. Then, possible malfunctions for each component were discussed. Finally, the subjects undertook a simulated ORS mission, during which envisioning of normal system response was practiced. Session 2 taught elementary diagnostic procedures such as checking a sensor bias or a valve leak. The subject then was required to plan testing procedures for five typical hypotheses. Each developed procedure was discussed by the experimenter until the subject developed (and understood) a correct procedure. Next, three real problems were given as exercises. Sessions 1 and 2 took 1.5 hours and 2 hours on the average, respectively.

The performance of the subject in the entire training sessions was closely monitored. The first session contained many questions to ascertain if the subject achieved proper understanding. The answers were checked during the same session and, whenever necessary, discussed again. Problem solving exercises were also attended by the experimenter and necessary discussion or re-explanation was provided. The result was that the initially poorer subjects would spend more time in training rather than end with poor understanding. By the end of the second session, all the subjects performed satisfactorily and showed little additional improvement in diagnostic skill.

The considerations which led to the design of experiment has been discussed in the overview section. The design for experiment 1 is shown in Figure 8. Each group was composed of three subjects and the Latin square was replicated three times using different problems.

Many different performance measures were tried with our data from the pilot experiment. The number of information gathering actions (#IGA) and the time to solve (Time) appeared to be appropriate performance measures. Although several other measures were examined with the data, they either turned out to have insufficient resolution or showed high correlations with the above measures. Thus, the above were the most important measures in this experiment. Time and #IGA showed virtually identical behavior both in the examination of aptness of the ANOVA model and tests of significance.

The data collected from 36 subject-problems were first analyzed to determine if there were significant interactions between problems and aiding levels. The interactions were found insignificant both in time ($p = .409$)

and #IGA ($p = .534$). This suggested that the interaction term can be excluded from the model and its sum of squares may be pooled with that of error term.

The Tukey test uncovered nonadditivity in the data of both Time and #IGA. The residual plot indicated that the cell standard deviations were proportional to cell averages. As this is frequently the case when the criterion is response time [Lee 1975, pp.291], a logarithmic transformation was suggested. After the transformation, the anomaly in the residual plot was fixed. The transformed data, both in Time and #IGA, appeared to adhere to the homogeneity and normality requirements for ANOVA better than the original scores. The interactions between aiding levels and problems were still insignificant. The Tukey test was performed again with the new scores and showed no significant nonadditivity.

The contribution of N aiding to both Time and #IGA was on the negative side, though not significant ($p = .096$ and $.381$, respectively). On the average, it corresponds to 31% increase in Time and 13% in #IGA.

These results may not simply be interpreted that N feature did not help the envisionment of normal system behavior or that the role of such envisionment in the diagnosis is unimportant. A proper interpretation may be that the normal envisionment could not be helped very well by providing external information because the process is too quick and deeply embedded in a larger cycle of human information processing. Another possibility is that envisioning normal system behavior was not a bottleneck in diagnostic performance.

We concluded the former interpretation was very likely considering the following. First, most subjects, after their main sessions, stated that the aid was not only uninformative, but also somewhat distracting or confusing. A subject said he wished he could get 'real' system behavior rather than 'normal' behavior. Second, the fairly strong negative aiding effects could not be explained if the aid helped only unimportant subtasks. Third, the negative aiding effect was notably stronger in Time than in #IGA. (This was the only occasion in which the two measures showed any notable difference in the analysis throughout experiment 1 and 2.) This implies that the aid forced the subjects to think for a longer time but did not greatly affect their diagnostic procedure. This result supported the subjects in reporting that the aid was confusing and distracting. Thus, we concluded that there was interference between N information and the operators' diagnostic information processing. Certainly, they do predict normal system behavior as a subtask: it is obviously necessary. But, when they seek information from the display, it was not of normal system behavior. This observation will be implemented in modeling of deep-reasoning diagnosis later in this paper.

Experiment 2

The second experiment was to assess the aiding effects of O and O-N features against unaided diagnosis. Nine new subjects, again industrial engineering students, were recruited as volunteers. Two training sessions which were virtually same as in the first experiment were given. In terms of content, the only difference was that the explanation of the new features replaced that of N feature. The design of experiment, shown in Figure 9, was also the same except for a different number of treatment levels and replication.

The procedure of statistical analysis was the same as in Experiment 1. First, the interactions between aiding levels and problems were found insignificant. After pooling the sum of squares for interactions into error sum of squares, the Tukey test for additivity was performed. No significant nonadditivity either in Time or #IGA was found. When the residual plots were examined, however, it was indicated that both measures needed to be logarithmically transformed. After the transformation, the new residual plots showed stabilized error variance. Again, the interactions between aiding levels and problems were insignificant. The Tukey test with the new scores yielded a much lower F value than before the transformation, confirming that the new scores fit the assumptions of the model better.

As results of the analysis of variance, both Time ($p = .0302$) and #IGA ($p = .0005$) showed significant effects of aiding. In Time, the improvement (i.e., decrease in Time) on the average was 34% by O aiding and 42% by O-N aiding. In #IGA, O aiding permitted 40% decrease while O-N aiding gave 44%. Neuman-Keuls tests were performed to determine if there were significant differences between pairs of aiding levels. Both O and O-N aiding levels had significantly different means when compared to the unaided mean. This result was identical for both Time and #IGA. In any measure, there was no conspicuous difference between O and O-N aiding.

The obvious conclusion is that both aiding features were effective in both measures and permitted solid enhancement of human diagnostic performance. In contrast to the N feature, these types of information appeared to be well accepted by the human process of diagnosis and helped the human in some important elements for his/her performance.

Experiment 3

The motivation for Experiment 3 was informal observation of subjects during Experiment 2. The effectiveness of O-N aiding in Experiment 2 appeared to decrease as the diagnosis proceeded. As is to be supported by more elaborate analysis later, this motivated us to investigate possible transitions between problem solving phases made by the diagnostician. Probably the most notable change in diagnosis as time passes was that the diagnostician began to deal with more explicit and individual hypotheses after the feasible hypothesis set size had been sufficiently reduced. In later phases with individual hypotheses, the characteristics of problem solving may be very different than the earlier phase of narrowing down the hypothesis set. Therefore, it was necessary to investigate the nature of diagnostic activity and proper form of aiding with such explicit hypotheses.

Due to its unique purpose, this experiment had an important difference in its setting from the first two experiments. In Experiments 1 and 2, the subjects solved whole diagnosis problems starting with primary symptoms. In the third experiment, the subjects determined whether a given hypothesis was true. At first, instead of being told of symptoms, the subject was allowed to perform some predetermined sensor readings which would indicate abnormal system behavior. Then, the subject was given a hypothesis to evaluate. Without needing to diagnose the real failure, the subject was to end the problem solving merely saying if he/she agreed at the hypothesis.

The effects of O-N aiding and O-H aiding were evaluated against unaided situations in two separate Latin square designs, i.e., Experiments 3-a and 3-b. They are shown in Figure 10. This was because, as mentioned earlier, it was not possible to assign both O-N and O-H aiding levels to the same subject due to expected interference. Although both Time and #IGA were collected, only Time was used in formal statistical analysis. Since the problems are much smaller in size than those of earlier experiments, #IGA is usually a small integer that would not easily lend itself to meaningful statistical analysis considering the vast difference in the subjects' diagnostic procedures. Otherwise, the analysis proceeded in a similar procedure as that of previous experiments.

In the analysis of the data from Experiment 3-a, the main question was what effects O-N aiding will have on the performance of diagnosis with a given hypothesis. First, the interactions between aiding levels and problems were tested and found insignificant ($p = .881$). Thus, a pooled error sum of squares were used for subsequent analysis. The Tukey test for additivity revealed the data were indeed additive. The residual plot also confirmed the model fitted the data quite well. It may be noted that, unlike the former experiments, no transformation was found necessary. The reason perhaps lies in the nature of the problems; these problems are just elementary subtasks which the operator should do numerous times in a whole diagnosis. As for the whole diagnosis time, the standard deviations were proportional to the means. That is, when a problem was more complex, the variation in the actual diagnosis time tended to be larger. This tendency most probably comes from the process of narrowing down the hypothesis set since the subtask of hypothesis testing did not show this property.

The performance was somewhat worse with O-N aiding than without it. Although not significant ($p = .192$), the difference on the average extended to 15.6 seconds (overall average was 67.4 seconds). The interpretation will be discussed with the evaluation of O-H aiding.

Experiment 3-b proceeded the same way except O-H aiding was tried in the place of O-N aiding. The interactions between aiding levels and problems were negligible ($p = .8593$). The additivity test confirmed that the data were additive. As in Experiment 3-a, the residual plot indicated that no transformation was needed. surprisingly, the effects of aiding appeared to be completely negligible (around 1 second, $p = .9546$).

The interpretation of these results is subtle. First, the O-N information was not relevant to the operator's activity to test a given hypothesis. The aid distracted the operator only to think about irrelevant information. This confirmed our earlier observation in Experiment 2 that the aiding effects of O-N information seemed to diminish as the diagnosis proceeds into its final stage. This observation, too, became a basis of our modeling of deep-reasoning diagnosis which is discussed in a later section.

Then, why was O-H aiding, which must be relevant to the given hypothesis, not effective? Two possibilities occur. First, the O-H information was simply not relevant to the problem solving. Otherwise, the information was relevant but trivial to the subjects. The first interpretation is not consistent with our previous results that, when irrelevant information was given to the subjects, the performance showed signs of degradation. The remaining choice is that the information, which is basically a set of suggestions for interesting observation, was already known to the subjects. That is, they already knew what to see even without the aiding; the aid only confirms it.

This interpretation could be further confirmed by a detailed process analysis. In Experiments 3-a and 3-b, 32 problems were solved without aiding. If O-H aiding had been provided with these problems, it would have suggested useful sensor reading actions 39 times. In 38 out of the 39 times (97.4%), the subjects collected equivalent information without it being suggested. Since they were ready to gather the O-H information whenever it was useful, the suggestions for this information by the computer were not able to improve the performance further. Because, unlike the O-N suggestions, O-H suggestions were just what the subjects were about to do, they were understood as trivial so that no performance decrement was caused by interference, either.

There was also an indication that the subjects planned valve operations and sensor readings together ahead of the actual operations. The subjects' collecting of O-H information was remarkably precise. There were 5 occasions in which the O-H aiding, if had been given, would have suggested uninformative readings. Failing in only one case out of 39 to look at useful O-H information, the subjects did not waste their time to do the uninformative sensor readings in any of the 5 occasions. Such precision may not be possible if the subjects were simply hunting around for useful observations by chance in scenes they just created. Most likely, the scenes were purposely planned aiming at the useful information. It should be noted that this tendency was unique and appeared only when an explicit hypothesis was given.

Summary

To summarize, O aiding and O-N aiding improved the diagnosis while N aiding did not. Actually, N aiding seemed to have negative effects. This suggests that the operator can effectively utilize O information, not N information, supplied from outside of his/her own information processing.

The usefulness of O-N aiding seemed to decrease over time perhaps as explicit hypotheses arose. In explicit hypothesis testing, O-N aiding showed a weak negative contribution while O-H aiding did not affect the performance at all. When weak negative effects were found, there seemed to be some interference caused by irrelevant information. On the other hand, O-H aiding was trivial and innocuous. The precision with which the subjects collected O-H information indicated that, when a hypothesis was given, the operational actions and data collection were usually planned together before the operations. This is an important observation in how the operators used their mental models.

A MODEL OF DEEP-REASONING DIAGNOSIS

Methodology

In this section, the experimental results will be integrated into a model of novel fault diagnosis.

The overall diagnostic procedure can be viewed as a combination of two elements: information processing tasks and a control strategy. Information processing tasks are subprocedures of diagnosis which can be characterized by their input, output and processes which take the input to produce the output. The control strategy is the way in which information processing tasks are selected.

The emphasis in this research has been on the information processing tasks, not the control strategy. There are several reasons for this. First, aiding novel fault diagnosis is the goal. Such diagnosis relies on causal reasoning about the system. To help causal reasoning, information processing tasks in which causal reasoning is embedded need to be understood. Second, we wanted to evaluate an aid which would be able to help the human to overcome cognitive limitations by some extent. While the aid would possess a similar causal reasoning capability to a human, it would not suffer the same cognitive limitations. This aid would be a more direct help to information processing tasks rather than the control strategy. Third, the findings from our research would permit insights to the structure of these information processes since our aiding approach was to provide various types of information which would substitute for the operator's information processing.

The emphasis on information processing led to a description of data flows rather than a flow chart. A flow chart would depict how the chronological sequence of various processes is controlled. In contrast, a data flow diagram would describe the necessary information input to a process, the expected output from a process, and the organization of processes through the links of information. This diagram helps to identify necessary subprocesses and alternative ways of automation.

A basic assumption connects our aiding experiments and the human information processing model: the human can better incorporate external information into his/her processing when the information becomes an alternative input to one of the higher level processes. An information processing task can be broken into processes, each of which can be broken into subprocesses. We assume that aiding information can be substituted for an entire process more effectively than for just an individual subprocess. There are several reasons to believe this assumption is reasonable. Because they are inner cycles in processes, subprocesses iterate and require input at higher rates. Also, the operator's working memory is more heavily loaded during a subprocess since the status of the higher level process, as well as that of the subprocess itself, should be retained. With the frequent cycles and heavy mental workload, it would be harder to perceive and apprehend externally supplied information [Wickens 1984, Rasmussen 1984].

As far as causal reasoning of the system operation is concerned, two directions of information processing should exist: observations to hypotheses and hypotheses to observations. The former task takes observations as input and produces hypotheses, while the latter starts from hypotheses and identifies necessary observations. Both tasks may be categorized as _search by evaluation_ according to Rasmussen's classification [Rasmussen 1984].

Observations to Hypotheses

This task is triggered by observations of system behavior and will be referred to as _data-driven search_. It occurs when the observations were collected without particular hypotheses or showed unexpected patterns that fell outside hypotheses of interest. It seemed therefore natural that the subjects performed this type of process more often in earlier phases of diagnosis. Since O-N aiding was useful in earlier phases, the information it supplied must be closely related to this task. The poor performance of N aiding, however, indicates that the human's use of N information is in a lower level subprocess, very likely to produce O-N information. Therefore, it is suggested that there is a process which filters the observations to pass only more interesting (i.e., unexpected) ones to the next process: N information is used for one of its subprocess. Obviously, there must be one more process to complete this task. In this second process, the human tries to come up with a set of plausible hypotheses that explain the observations.

Some of the interesting observations may be remembered to evaluate future hypotheses throughout the diagnosis. The above constraints allow one to conceive a model of the data-driven search as represented in Figure 11.

Two processes were identified. The first process is filtering observations. Only the observations which passed this filtering are used in the following process of entertaining hypotheses. The filtering process contains a reference mental model of the system. The reference model is a mental model that produces standard behavior against which observed system behavior is continuously compared and judged as expected or unexpected. At first, the reference model behavior is that of normal system. As more observations are accumulated, however, some abnormal system behavior would also become expected even though the reason may not be understood. An expected observation does not carry additional information and should be filtered out as trivial. Thus, the reference model should evolve incorporating more and more observations of actual system behavior. Converging to the actual system in its behavior, the reference model would lower the probability of unexpected observations. Consequently, the efficiency of unplanned observations would decrease and the data-driven search would become less useful as the diagnosis proceeds.

In earlier phases of diagnosis, when the reference model behavior is normal, O-N aiding replaces the whole filtering process and provides input information to the hypotheses entertaining process. According to our basic principle, it should be easier for the human to incorporate such information into his information processing. This was supported by the experimental result that O-N aiding improved the diagnostic performance. However, the gradual departure of the reference model from normal system behavior would degrade the relevance of O-N aiding in the filtering. It was supported by the observation that O-N aiding was mostly useful in earlier phases of problem solving.

O aiding enhanced the observations which are input to the filtering process. The enhancement is in fact presentation of observed system behavior at a higher level of abstraction than the sensor displays [Rasmussen 1984]. For example, while the operator would normally look at individual pressure points to check the system behavior, O aiding would display a mass flow which is not the behavior of a component, but of a path. Since this level, being more functional, allowed more appropriate information coding for the operator's use, it should improve the filtering process. The experimental results supported this.

The prediction of normal system behavior (N aiding) is at first equivalent to the subprocess of running the reference model. This activity is internal to the filtering process, neither replacing a process nor providing better information to a process. As a result, there may be little chance to improve human diagnosis by providing this information from outside. Actually, the experiment showed that N aiding had rather negative effects, though not significant, perhaps due to distraction.

Hypotheses to Observations

When given hypotheses are to be evaluated, the operator would build a testing plan that may prove one hypothesis and disprove the rest. This task is called hypothesis-driven search. Experiment 3-a indicated that, by demonstrating poor performance of O-N aiding, this task was very different from data-driven search in its information processing.

This type of process tends to be employed more often toward the final stage of diagnosis as the data-driven search loses its efficiency. An important restriction of this process is that the hypothesis should be sufficiently explicit for the diagnostician to perform mental simulation based on it. There are usually too many explicit hypotheses that are feasible in earlier phases of diagnosis. Therefore, the data-driven search may be preferred in narrowing down the feasible hypothesis set. Toward the end of diagnosis, however, the number of feasible hypotheses would become smaller and the need of testing the remaining hypotheses individually would increase. Then, the hypothesis-driven search dominates the diagnosis.

In Experiment 3-b, we forced the subject to perform this process by assigning a hypothesis to test. The experimental result that O-H aiding did not improve the human diagnosis can be explained in this model. O-H aiding suggested sensor readings which would show the difference between actual and hypothetical system behavior. When the hypothesis is false, a right test would reveal the existence of O-H behavior to disprove the hypothesis. Thus, O-H information is certainly relevant to the hypothesis testing. It is reasonable to expect O-H aiding to be helpful if the operator collects observations and filters them as in the data-driven search. If, however, the tests are planned by predicting observable differences (as in Figure 12) depending on whether the hypothesis is true, O-H information is identified before the actual testing operation. In this case, externally suggested O-H information would only be redundant and would not improve the performance.

The latter case was supported by the experiment; the aid gave no performance improvement; the operators collected O-H information in an extremely efficient manner even in unaided diagnoses, in which they were not given the suggestions by the aid. Therefore, it is safe to conclude that the operator, when a hypothesis is given, runs his/her mental model to determine a test that would distinguish the given hypothesis from other hypotheses. Figure 12 describes the model of this task.

<u>Control Strategy</u>

The control strategy is both highly dynamic and individualistic. Operators switch frequently between information processing tasks. The selection of tasks depends on the assessment of relative efficiency and effectiveness of different tasks in different situations. For example, if the diagnostician is equipped with very inexpensive testing methods to check every component directly, the cost of hypothesis-driven search will be drastically reduced from what it is in the ORS diagnosis. This observation suggests the possibilty that the control strategy can be changed when aiding affects the efficiency of elementary tasks.

Although the two information processing tasks are the most important elements, the strategy may involve other types of information processing. Topographic search [Rasmussen 1984] can be used either to entertain hypotheses or the necessary observations for a hypothesis. In fact, this is believed to be the frequent way in which the operator, when performing data-driven search, selected the data to begin with.

Regarding the control strategy, the only observation we could be assured of was that the subjects gradually transitioned from data-driven to hypothesis-driven search as the diagnosis proceeded. This was perhaps because the reduction of the size of feasible hypothesis set changed the relative efficiency of two processes. For instance, with only one hypothesis to deal with, explicit planning of test by hypothesis-driven search must be more efficient. It may also be partly because, as we have already discussed, the data-driven search lost its efficiency as observations were accumulated.

As a conclusion, the detailed modeling of information processing tasks helped to integrate our findings and observations of human operator's novel fault diagnosis. The models of human information processing tasks were useful in explaining the aiding effects of various types of information. It should also be useful to predict effects of aiding to be proposed in the future. Such predictions, in turn, may be tested in experiments to verify the model.

CONCLUSION

An aiding approach has been described and evaluated for novel fault diagnosis in complex systems. To the best of our knowledge, this approach is unique in the following ways. First, the emphasis is on novel rather than routine faults. Second, it contains a qualitative model that may correspond to the human's internal model of the system. This model represents knowledge only of how the system behaves. Therefore, this aiding approach does not rely on proceduralized knowledge. Third, the qualitative model is the basis for much of the aiding that takes place.

The experimental results confirmed that a deep-reasoning diagnosis can be aided, without disturbing the human diagnostic procedure, by providing relevant information. However, the results also suggested that the aiding information should be compatible with the human information processing. This emphasizes the importance of understanding the human information processing to build an effective aid. A principle of particular importance is that the information from/to higher-level processes is better incorporated into the human's information processing. The findings and observations were integrated into an effort to model the information processing tasks for deep-reasoning diagnosis.

ACKNOWLEDGMENT

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

REFERENCES

Davis, R., "Diagnostic reasoning based on structure and behavior," Artificial Intelligence, Vol. 24, pp. 347-410, 1984.

De Kleer, J. and Brown, J.S., "Assumptions and ambiguities in mechanistic mental models," in D. Gentner and A.L. Stevens (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Edwards, A.L., Experimental Design in Psychological Research, New York, NY: Holt, Reinhart and Winston, 1972.

Forbus, K., "Qualitative process theory," Artificial Intelligence, Vol. 24, pp. 85-168, 1984.

Gentner, D. and Stevens, A.L. (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Greenstein, J.S., "The use of models of human decision making to enhance human-computer interaction", Proc. of 1980 IEEE Int'l Conference on

Cybernetics and Society, Cambridge, October, pp. 968-970.

Kuipers, B., "Commonsense reasoning about causality: Deriving behavior from structure," *Artificial Intelligence*, Vol. 24, pp. 169-203, 1984.

Lee, W., *Experimental Design and Analysis*, San Francisco, CA: Freeman, 1975.

Mehle, T., "Hypothesis generation in an automotive malfunction inference task," *Acta Psychologica*, Vol. 52, pp. 87-106, 1982.

Miller, R.A., Pople, H.E. Jr., and Myers, J.D., "INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine," in *Readings in Medical Artificial Intelligence*, Clancey, W.J. and Shortliffe, E.H. (eds.), Reading, MA: Addison-Wesley, 1984.

Morris, N.M. and Rouse, W.B., "The effects of type of knowledge upon human problem solving in a process control task," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-15, No. 6, 1985.

Morris, N.M. and Rouse, W.B., "Review and evaluation of empirical research in troubleshooting," *Human Factors*, Vol. 27, No. 5, October 1985.

NASA Johnson Space Center, "Orbital refueling demonstration system description," Program Development Office, October 21, 1985.

Neter, J. and Wasserman, W., *Applied Linear Statistical Models*, Homewood, IL: Irwin, 1974.

Newell, A. and Simon, H.A., *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall, 1972.

Wohl, J.G. "Cognitive capability versus system complexity in electronic maintenance," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-13, No. 4, 1983.

Yoon, W.C. and Hammer, J.M. "Aiding the operator during novel fault diagnosis," to appear in IEEE Trans. Systems, Man, and Cybernetics, 1987.

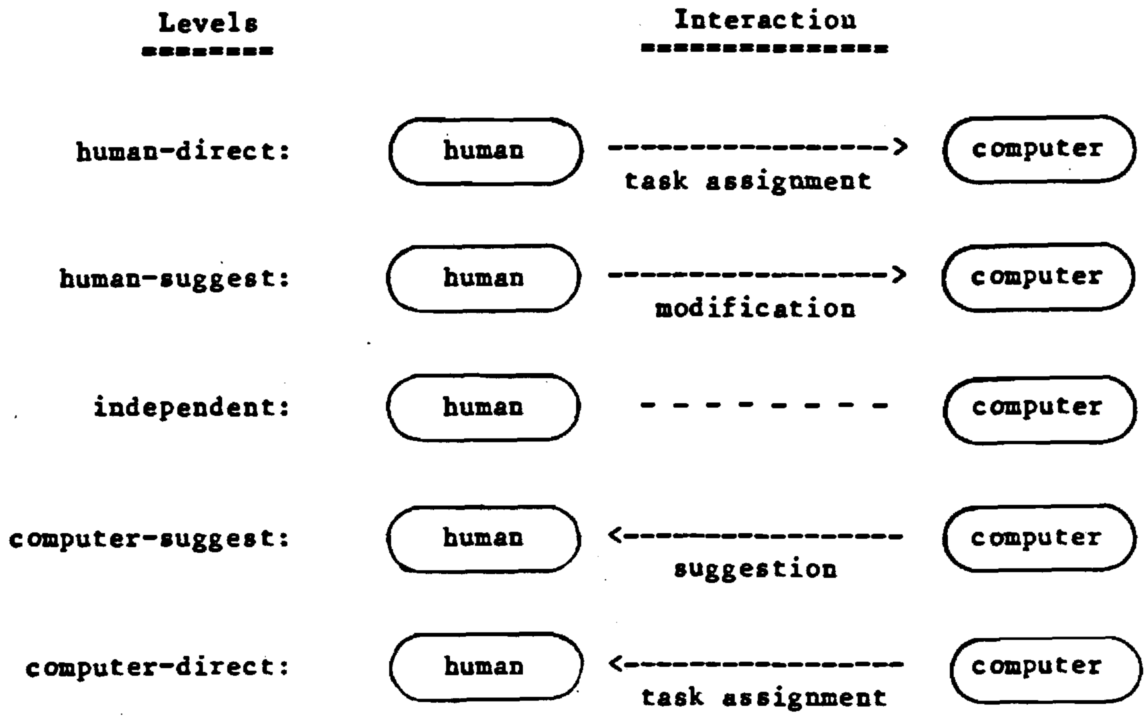


Figure 1. Levels of Interaction

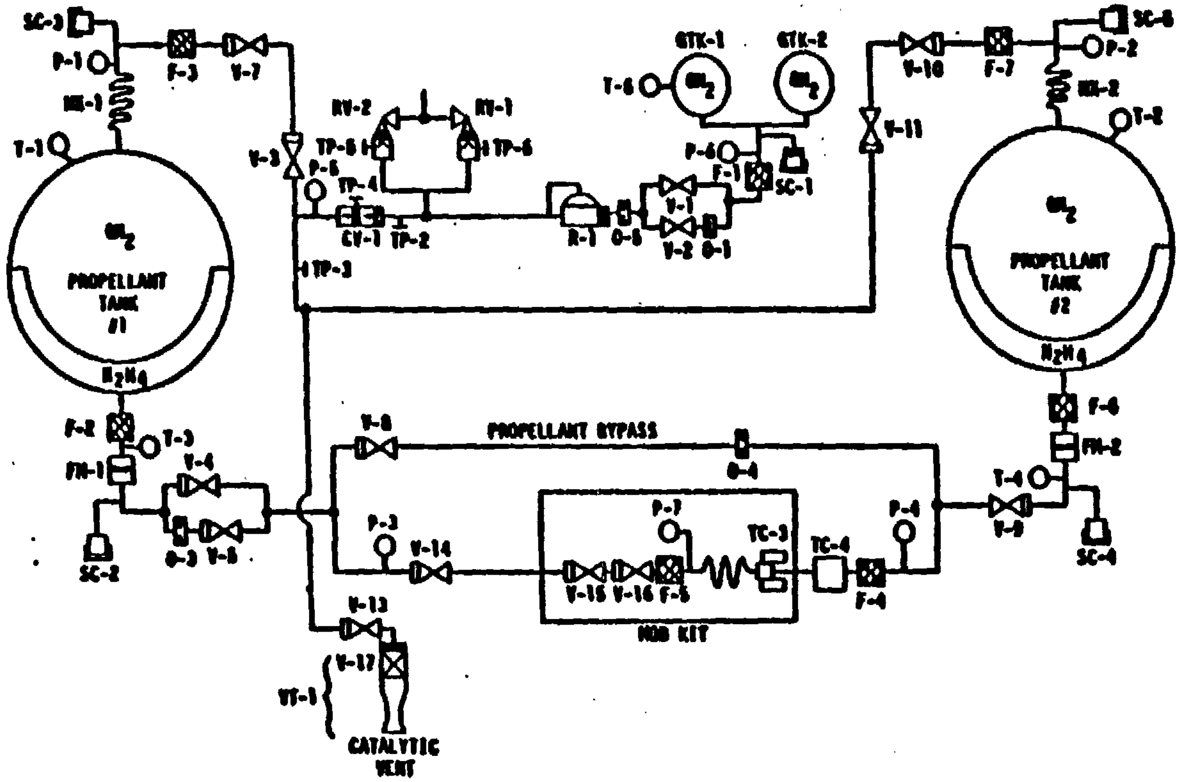


Figure 2. The Orbital Refueling System.

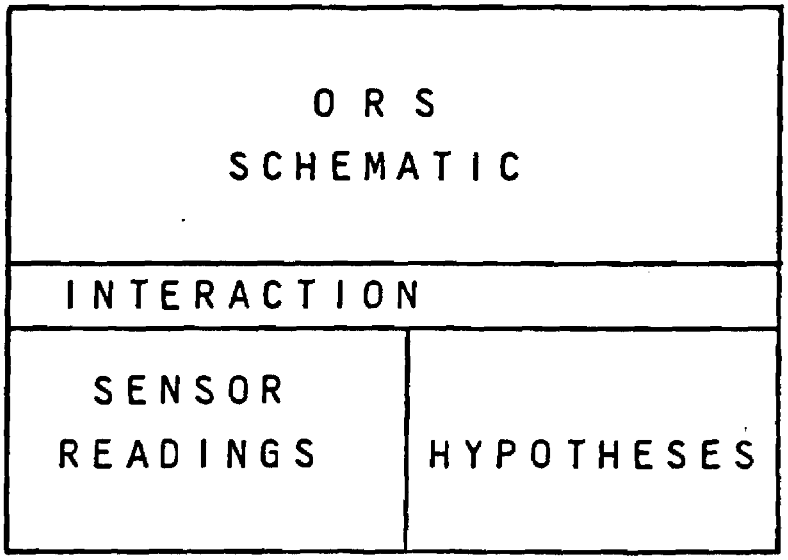


Figure 3. The operator's display.

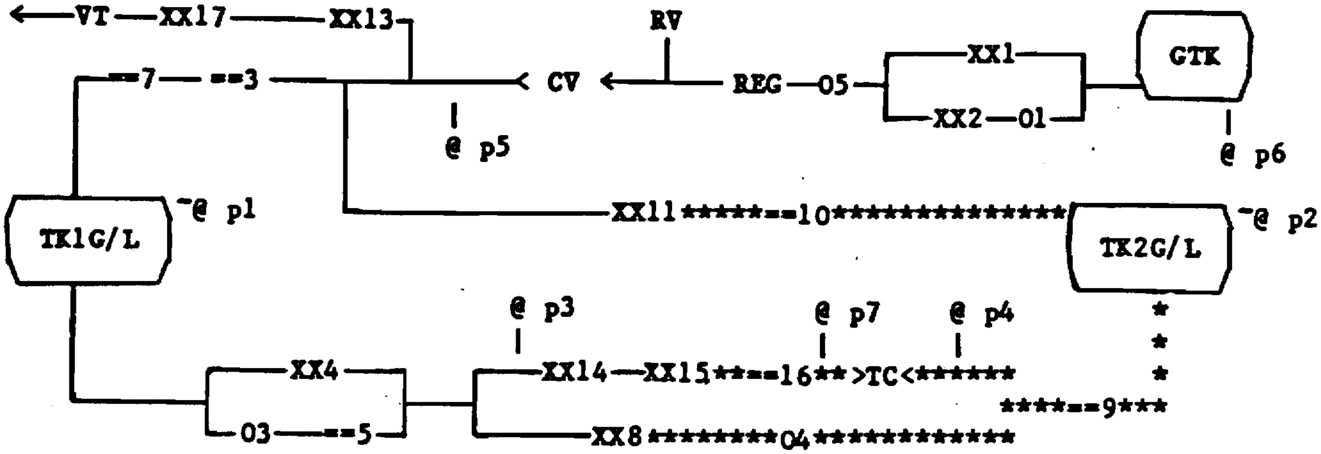


Figure 4. The normal response (N).

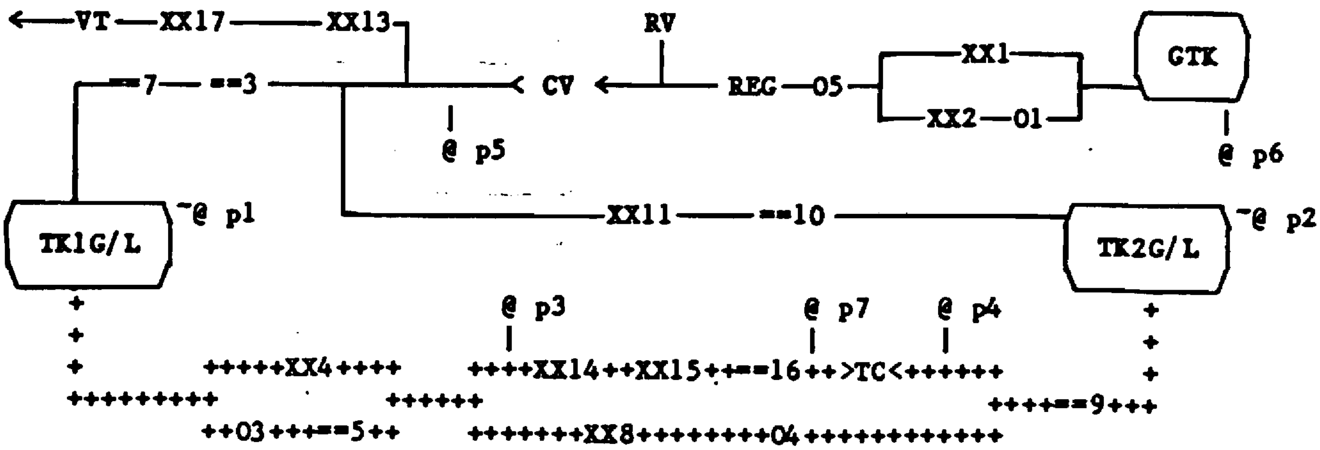
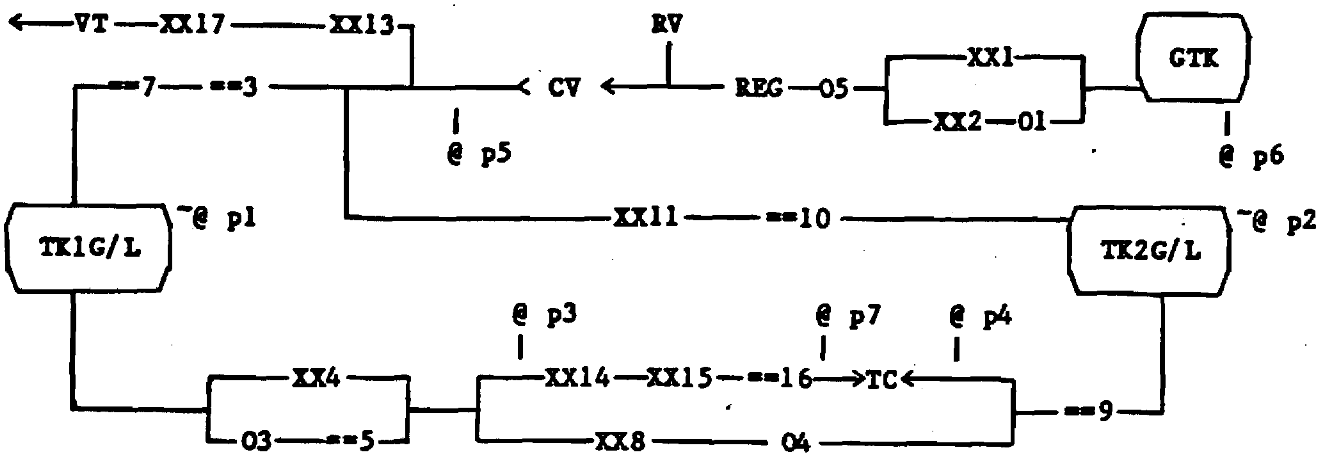
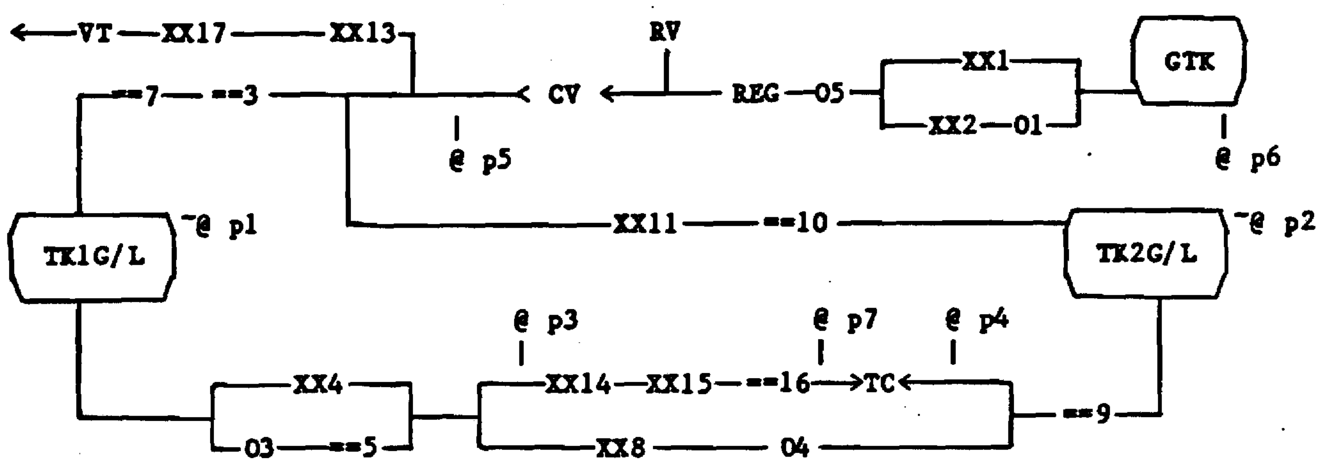


Figure 5. The observed response (O).



SEE
* D P1
* D P2
* C P2 P4

Figure 6. Deviation from normal behavior (O-N).



SEE
* D P1
* C P2 P4

(L V10): leak

Figure 7. Deviation from hypothesized behavior (O-H).

PROBLEMS

		P1	P2	P3	P4	P5	P6
GROUPS	G1 (S1-S3)	-	N	-	N	-	N
	G2 (S4-S6)	N	-	N	-	N	-

where N: N-aided situation
-: unaided situation

Figure 8. Latin Square Design for N effects in Experiment 1.

PROBLEMS

		P1	P2	P3	P4	P5	P6
GROUPS	G1 (S1-S3)	-	O	O-N	-	O-N	O
	G2 (S4-S6)	O	O-N	-	O	-	O-N
	G3 (S7-S9)	O-N	-	O	O-N	O	-

where O: O-aided situation
O-N: O-N aided situation
-: unaided situation

Figure 9. Latin Square Design for O and O-N in Experiment 2.

PROBLEMS

		P1	P2	P3	P4	P5	P6	P7	P8
GROUPS	G1(S1,S2)	-	A	-	A	-	A	-	A
	G2(S3,S4)	A	-	A	-	A	-	A	-

where A: aided situation (O-N or O-H)
-: unaided situation

Figure 10. Latin Square Designs for O-N and O-H effects in Experiment 3.

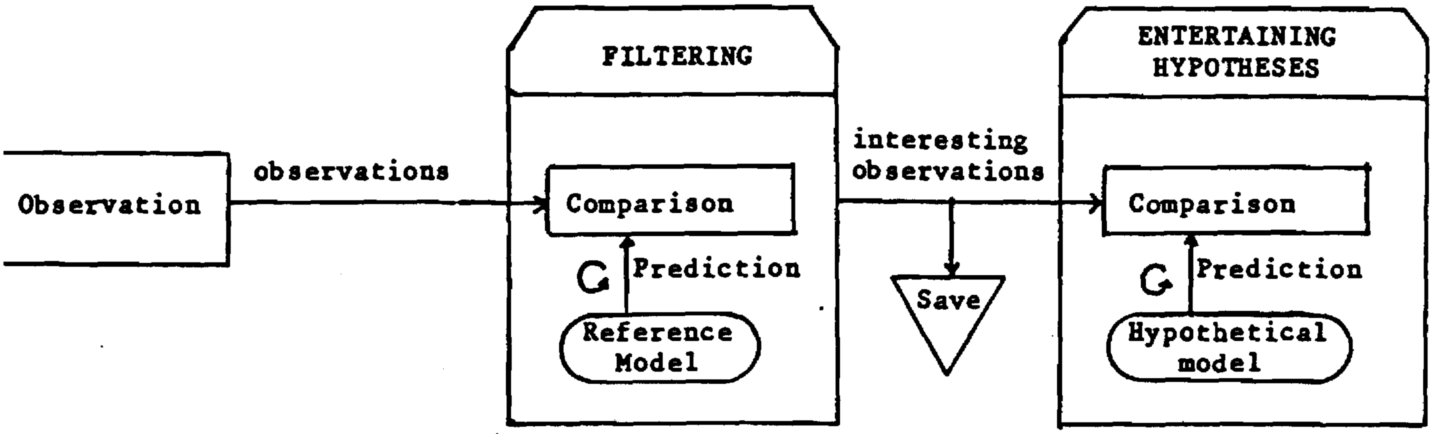


Figure 11. Data-driven Search

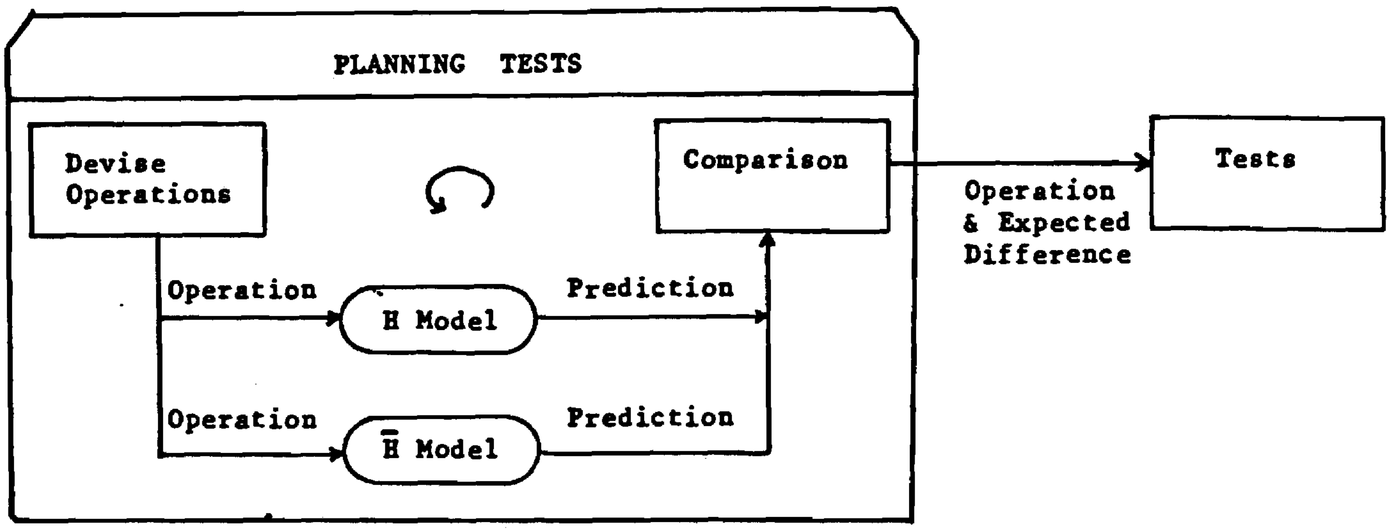


Figure 12. Hypothesis-driven Search